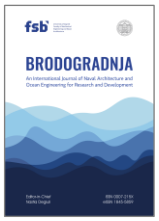
Brodogradnja Volume 77 Number 4 (2026) 77402

University of Zagreb
Faculty of Mechanical
Engineering and Naval
Architecture

journal homepage: www.brodogradnja.fsb.hr

Brodogradnja

An International Journal of Naval Architecture and
Ocean Engineering for Research and Development



Research on dynamic threshold warning and MLOps application for marine diesel engine exhaust gas temperature

Zhaoyi Wang[1], *Huibing Gan*[1,*], *Zhennan Jin*[2], *Zhibo Lei*[1]

[1]*Marine Engineering College, Dalian Maritime University, Dalian 116026, China*
[2]*COSCO SHIPPING Heavy Industry (DALIAN) Co.,Ltd.*

ARTICLE INFO

Keywords:
Marine diesel engine
Interval-specific modelling
LightGBM
Dynamic threshold
MLOps

ABSTRACT

To address the challenges of inaccurate exhaust gas temperature (EGT) prediction under varying load conditions and the difficulty in engineering implementation within intelligent marine engine rooms, this study proposes a comprehensive solution. The load (L), turbocharger speed (N), and turbocharger exhaust outlet temperature (T1) are selected as core input parameters using Spearman's rank correlation analysis, reducing the number of parameters by approximately 60 % compared to multi-parameter models. The 0-80 % load range is equally divided into eight intervals of 10 % each, with a dedicated Light Gradient Boosting Machine (LightGBM) predictor developed for each interval. This approach achieves a test set Mean Absolute Error (*MAE*) of 1.8584, Mean Squared Error (*MSE*) of 9.7828 and coefficient of determination (R^2) of 0.9866, significantly outperforming a full-interval model. And based on the data characteristics of each subinterval, a dynamic threshold warning system with three-level logic is developed, which achieves a fault identification rate of over 92 % for simulated faults. Furthermore, a four-layer Machine Learning Operations (MLOps) architecture is implemented, with the model containerized into a 480 megabytes (MB) image and deployed via a local automated pipeline suitable for network-limited, low-power edge environments. System resource usage remains below 650 % Central Processing Unit (CPU) and 1900 mebibytes (MiB) memory. Validation through six-hour offline closed loop operation and a 180-day simulated aging test confirms the solution's robustness and practical potential for real-ship intelligent engine room health management.

.

1. Introduction

In response to the International Maritime Organization (IMO)'s ambitious greenhouse gas (GHG) emission reduction strategy, which strives for net-zero GHG emissions for international shipping by around 2050 [1], the maritime sector is experiencing a comprehensive transition towards green [2] and intelligent operations [3]. This transition has elevated emission control and navigation safety to the status of core mandatory demands for the entire industry [4]. As the core carrier of ship intelligence, the intelligent engine

* Corresponding author.
E-mail address: ghbzq@dlmu.edu.cn

https://doi.org/10.21278/brod77402
Received 26 December 2025; Received in revised form 13 March 2026; Accepted 20 March 2026
Available online 31 March 2026
ISSN 0007-215X; eISSN 1845-5859

room imposes higher requirements on the health management of key equipment. Marine diesel engines, the core energy source of intelligent engine rooms, have operational reliability that directly determines ship power supply and navigation safety,and their combustion efficiency is closely linked to the industry's decarbonization goals [5]. Exhaust gas temperature (EGT) of diesel engines, as a key indicator reflecting combustion efficiency and in-cylinder thermodynamic state [6], is directly associated with core operational links including fuel injection, air supply and combustion reaction, and is critical for controlling fuel consumption and reducing carbon dioxide emissions [7]. Most faults of diesel engines and their core auxiliary systems are transmitted to EGT through changes in combustion efficiency, causing abnormal fluctuations [8]. Such fluctuations often precede visible equipment failures and serve as sensitive signals for capturing early fault symptoms [9]. For example, turbocharger blade fouling leads to insufficient air intake and incomplete combustion, resulting in a gradual abnormal rise in EGT. Without timely warning and handling, it may further induce severe faults such as cylinder overheating and turbine failure, endangering navigation safety [10] and exacerbating unnecessary emissions. Therefore, accurate capture of EGT abnormal characteristics of marine diesel engines can precisely assess the operational state of the engine's core systems, enable early fault identification and prediction [11], and reduce fault incidence.

Traditional monitoring methods for marine diesel engine status rely on manual periodic recording and empirical judgment, which suffer from delayed data collection, easy omission of early faults and subjective bias, making it difficult to meet the real-time and accurate monitoring needs of intelligent engine rooms [12]. To mitigate these limitations, machine learning has become the core enabling technology for health management of intelligent engine rooms [13]. Especially in the field of marine diesel engine prediction research, machine learning algorithms exhibit significant advantages with their powerful nonlinear fitting and data feature mining capabilities, and are widely used in fault feature extraction, operational status evaluation, abnormal trend prediction and warning [14]. Different machine learning frameworks feature distinct characteristics, enabling personalized adaptation to the diverse operating scenarios, data types and monitoring requirements of marine diesel engines, and become the core technical support for promoting the intelligent upgrading of marine diesel engine health management [15].

To achieve early identification and prediction of diesel engine faults [16], the academic community has conducted extensive exploration and research. Liu et al. [17] constructed a CNN-BiGRU prediction model, utilizing CNN to extract multi-dimensional parameter features and BiGRU to mine temporal dependencies, while setting dual warning thresholds to realize rapid identification and early warning of EGT abnormalities—this work achieved a solid prediction accuracy foundation for the field but failed to account for load-specific operational differences. Ji et al. [18] built a CNN-BiLSTM-Attention hybrid model, innovatively adopting Mahalanobis distance combined with the 3σ criterion to construct a health index model and optimizing the fault warning implementation path from both architectural optimization and warning method innovation; this model improved prediction accuracy compared to prior work, yet its threshold design lacked adaptability to variable load conditions. Tian et al. [19] addressed the pain point of model hyperparameter optimization by integrating CNN's spatial feature extraction capability with BiLSTM's temporal analysis ability and incorporating a Particle Swarm Optimization (PSO) step to eliminate inefficient manual tuning—this approach further enhanced prediction precision while achieving a breakthrough in model efficiency, but the increased model complexity resulted in relatively slow inference speed. Sun et al. [20] proposed a CNN-LSTM-MLA model targeting the adaptability challenge of diesel engines under complex operating conditions, deepening the dimension of feature fusion through multi-layer attention mechanisms and validating the model's highest prediction accuracy and strong robustness among these studies under multiple sailing conditions; however, the multi-layer attention structure significantly increased computational overhead, hindering engineering deployment. These studies have sequentially advanced EGT prediction accuracy and innovated warning methods, yet they have resulted in three unresolved technical gaps that remain unaddressed today:

(a) Single models used for all load conditions fail to adapt to operational differences, hindering accurate capture of temperature variation patterns across loads.

(b) Thresholds insufficiently account for parameter characteristic differences under varying loads, leading to inadequate adaptability and high risks of false alarms or missed detections during condition transitions.

(c) Overemphasis on prediction accuracy, pursued at the cost of escalating model complexity and reduced inference efficiency, slows down inference speeds, fails to meet real-time monitoring demands in actual ship scenarios, and ultimately results in insufficient engineering implementability and difficult deployment of warning systems.

Mature research from other fields offers targeted insights to address these gaps. These studies, though in different research fields, share the same core demands with marine diesel engine EGT monitoring. Zhao et al. [21] proposed the Quality-related Stepwise Sequential Phase Partitioning (QSSPP) algorithm in manufacturing, dividing production processes into phases with dedicated regression models—validating interval-based modeling advantages and providing a methodological reference for diesel engine load interval division. Karimi et al. [22] introduced the Shift-ACT framework for domain generalization, using class-specific dynamic thresholds to adapt to distribution shifts, which lays a technical foundation for designing load interval-specific EGT residual dynamic thresholds. Oyucu and Aksöz [23] selected LightGBM with low computational overhead and high accuracy for time series data and MLOps ensuring engineering consistency and scalability [24] for wind power prediction, achieving low-latency inference and demonstrating a mature paradigm for model selection and deployment.

To address the existing shortcomings in the research on the exhaust gas temperature of marine diesel engines, this study draws on mature technical ideas from other related fields and proposes a comprehensive and integrated solution that can simultaneously resolve the three aforementioned unsolved technical gaps. First, this study screens core parameters via rank correlation analysis and designs a lightweight, interpretable feature system adapted to the characteristics of exhaust gas temperature. It then innovatively develops interval-specific LightGBM models for different load ranges to improve the prediction accuracy under variable load conditions. Furthermore, based on the unique data characteristics of each load subinterval, the study designs a three-level logic dynamic threshold warning system to balance the sensitivity and stability of early warning. Meanwhile, it designs and constructs a four-layer MLOps architecture optimized for the resource constraints of shipboard edge environments, enabling model containerization and lightweight automated deployment. Ultimately, multi-dimensional tests verify that the solution exhibits favorable robust performance and engineering practicality.This study aims to bridge the gap between EGT warning technology and engineering application, improve the accuracy of EGT prediction under variable load conditions and the feasibility of the warning system, and provide a reliable and deployable technical scheme for the early fault identification and health management of marine diesel engines in real-ship intelligent engine rooms, thus supporting the digital and intelligent management of ship intelligent engine rooms.

2. Related Theoretical and Technical Foundations

2.1 Core Principles of the LightGBM Regression Model

LightGBM is an efficient ensemble learning engineering implementation framework improved based on the Gradient Boosting Decision Tree (GBDT) algorithm [25]. Its core advantage lies in balancing training efficiency and prediction accuracy through engineering innovation mechanisms (such as efficient node splitting strategies and parallel computing architecture), making it particularly suitable for large-scale data processing needs in industrial scenarios. It is widely used in various regression [26] and classification tasks [27]. Figure 1 shows the core process of the LightGBM regression model.

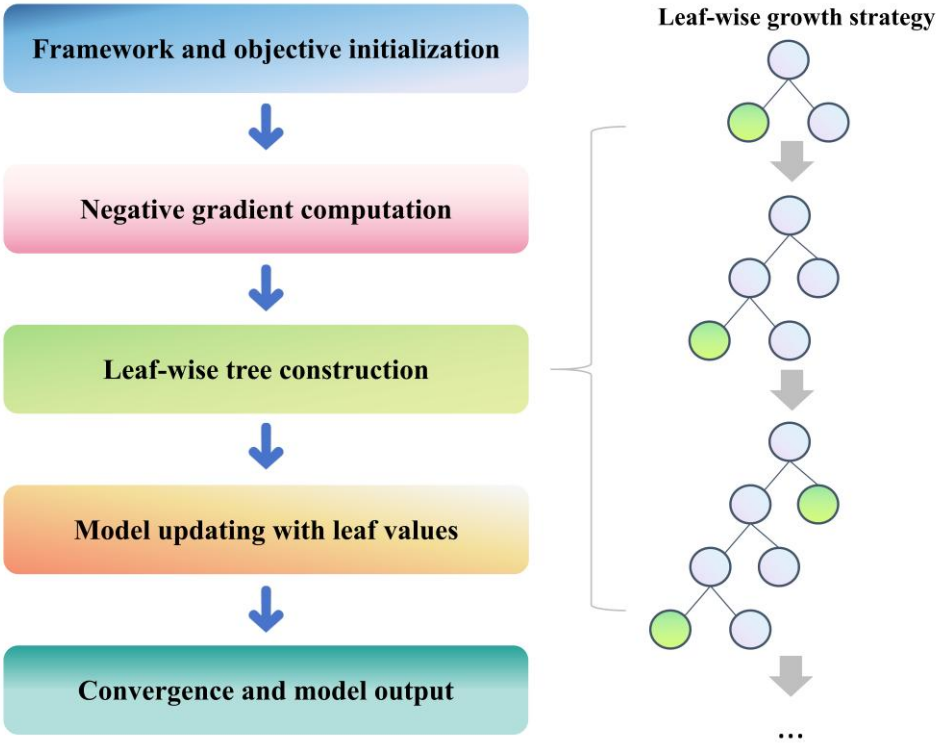


Fig. 1 Core flowchart of the LightGBM regression model

In terms of tree growth strategy, LightGBM adopts a leaf-wise mode, which differs from the traditional GBDT's level-wise strategy [28]. Its core logic is to select only the leaf node with the greatest information gain for splitting in each iteration, rather than expanding all layer nodes one by one. Information gain is calculated based on node entropy. The formula for Information Gain [29] is shown in (1).

$$Gain = H_{parent} - \left(\frac{n_L}{n_{parent}}\right) H_L - \left(\frac{n_R}{n_{parent}}\right) H_R \tag{1}$$

where H_{parent}, H_L, and H_R are the entropy values of the parent node, left child node, and right child node, respectively; n_L and n_R are the sample numbers of the left and right child nodes; n_{parent} is the sample number of the parent node. This strategy allows the model to prioritize deep optimization of high-gain regions, capturing finer-grained feature patterns with the same number of iterations, while preventing overfitting by limiting the maximum tree depth and minimum leaf samples, adapting to the complex distribution characteristics of industrial data.

Furthermore, LightGBM supports multiple loss functions to adapt to different regression needs [30]. Mean Absolute Error (*MAE*) is robust to outliers, suitable for scenarios with abnormal data fluctuations, and its specific calculation formula [31] is shown in (2).

$$MAE = \left(\frac{1}{n}\right) \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2}$$

Mean Squared Error (*MSE*) is more sensitive to detailed data changes, accurately reflecting the deviation between predicted and actual values. Its specific calculation formula [32] is shown in (3).

$$MSE = \left(\frac{1}{n}\right) \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3}$$

In the above two formulas, y_i represents the actual value of the sample, $\hat{y}_i$ represents the model's predicted value, and n represents the total number of samples.

The goodness of fit of the model is evaluated by the coefficient of determination (R^2) . Its core logic is to measure the model's ability to explain data variation. The specific calculation formula for R^2 [33] is shown in Equation (4).

$$R^2 = 1 - \frac{SSR}{SST} = 1 - \frac{[\sum_{i=1}^{n}(y_i - \hat{y}_i)^2]}{[\sum_{i=1}^{n}(y_i - \bar{y})^2]} \tag{4}$$

where $\bar{y}$ is the mean of the sample actual values. R^2 ranges from $R^2 \in [0,1]$. A value closer to 1 indicates a better fit of the model to the data patterns and higher reliability of the prediction results.

LightGBM also integrates multiple regularization mechanisms to further suppress overfitting, improving large-scale data processing efficiency through feature and data parallel computing [34]. For categorical features, it can directly process them by sorting based on gradient means and finding the optimal grouping threshold, avoiding dimensionality explosion while preserving the intrinsic relationships of categorical features. These characteristics collectively give LightGBM high efficiency and accuracy in high-dimensional, large-scale time series data scenarios [35], making it one of the preferred frameworks in the field of industrial modeling.

2.2 Core Technologies of MLOps

MLOps (Machine Learning Operations) is a technical system that supports the full lifecycle management of models in offline, edge hardware-constrained scenarios [36]. Its core lies in dividing the process into three stages: Machine Learning (ML), Development (Dev), and Operations (Ops). Through automated collaboration, it ensures stable operation and continuous adaptation of models in harsh environments, featuring localization, low consumption, and easy operation and maintenance [37]. Figure 2 illustrates the core process based on these three stages.

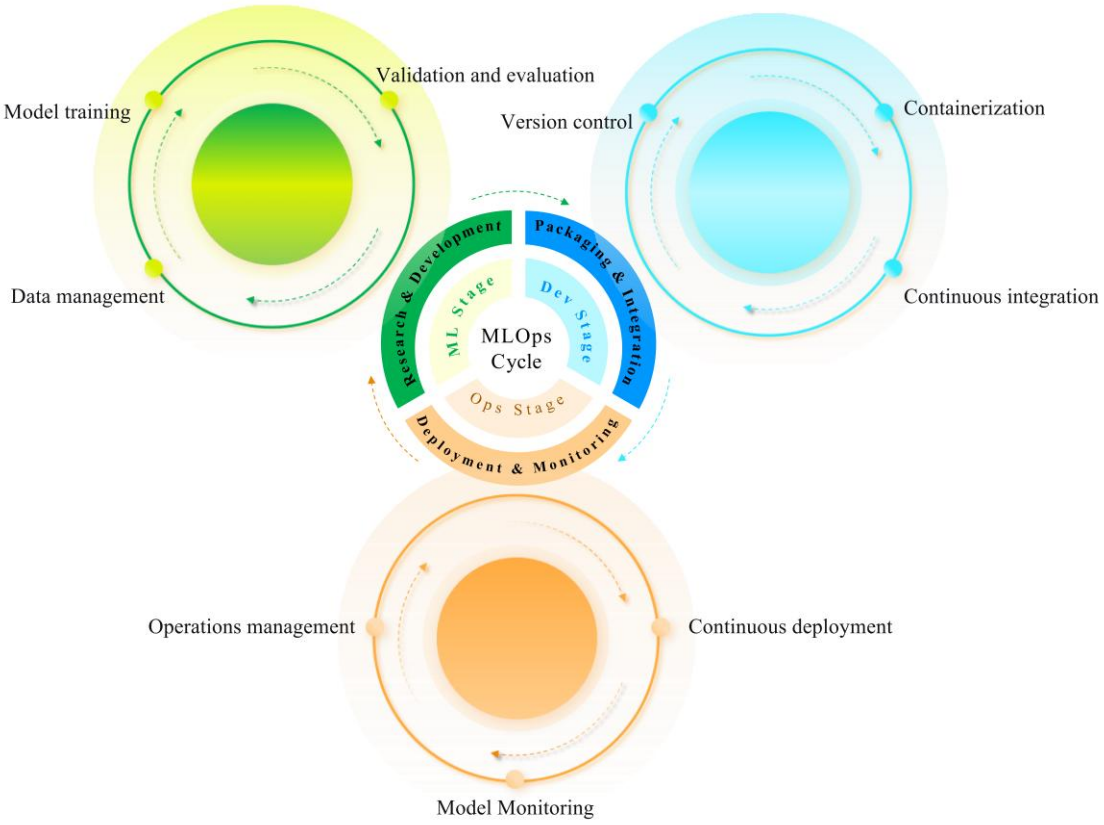


Fig. 2 Core flowchart of MLOps

The ML stage focuses on model development and validation and is the decision-making starting point of the process [38]. Under offline constraints, the system relies entirely on local closed loops for automated data acquisition, preprocessing, and version management to ensure input consistency. When monitoring detects that model performance (e.g., prediction accuracy) falls below a preset threshold, a retraining process including hyperparameter optimization is automatically triggered, generating candidate models with version tags. New models must pass strict evaluation on the local validation set to proceed to the next stage; otherwise, the experimental iteration restarts. The entire process data and parameters are recorded to form a traceable knowledge base.

The Dev stage is responsible for engineering the validated model prototype into a standard product, achieving the transition from experiment to production. Through strict code and model version control (e.g., Git) and using Docker multi-stage build technology to remove redundant components, lightweight, environment-consistent container images are packaged, adapting to edge device resource limitations. Simultaneously, a Continuous Integration (CI) automated pipeline is built to automatically trigger testing and verification upon code or configuration updates, ensuring reliable, reproducible deployment packages [39].

The Ops stage aims to complete the safe deployment and long-term stable operation of the model. Through a Continuous Deployment (CD) mechanism, images are automatically deployed to edge devices, supporting canary releases and one-click rollback to ensure business continuity. The system monitors hardware resources, inference latency, and business metrics in real-time. Once data drift or performance degradation exceeds limits, local alerts are triggered immediately [40]. For non-professional operation and maintenance scenarios, a visual interface is provided to simplify operations such as version management and parameter adjustment, and data encryption and backup mechanisms are integrated for comprehensive risk control.

MLOps, through the seamless connection and automated closed loop of the ML, Dev, and Ops stages, provides key technical support for the large-scale deployment and long-term operation of models in "offline, low-computing-power" industrial edge scenarios [41].

3. Core Development Process of the Warning System

3.1 Data Processing

The "Xin Hong Zhuan" is an intelligent vessel integrating research and training functions [42]. As a mobile experimental platform for testing and validating intelligent ship technology at sea, it possesses capabilities for autonomous navigation, remote control, and independent operation. Regarding its propulsion system, it is equipped with two fully electric azimuth thrusters (each rated at 1500 kW), powered by three 1520 kW diesel generating sets.

This study selects the diesel engine of the ship's No. 1 main diesel generating set (specifically, a Wärtsilä 8L20 model) as the primary research object. Its technical parameters are shown in Table 1.

Table 1 Technical parameter table for the diesel engine of No. 1 main diesel generating set

Engine Parameter	Unit	Value
Cylinder bore	mm	200
Number of cylinders	/	8
Rated power	kW	1600
Rated speed	rpm	75000
Piston displacement per cylinder	L	8.80
Stroke length	mm	280
Clockwise firing order	/	1-3-7-4-8-6-2-5

Data is collected from multiple sensors on the "Xin Hong Zhuan" ship, covering scenarios from economical speed and partial load to rated load conditions. Sufficient data is extracted for each load condition interval to ensure the adequacy of subsequent model training, the reliability of prediction results, and the generalizability of research conclusions. The diesel engine of the ship's No. 1 main diesel generating set (hereinafter referred to as the diesel engine) comprises 8 cylinders. This study uses the average EGT of the diesel engine's 8 cylinders recorded under normal operating conditions as the target variable for prediction analysis.

Combining relevant literature in the marine diesel engine field and technical documentation provided by the diesel engine manufacturer, 7 potential influencing parameters that may be correlated with diesel engine EGT are initially selected as candidate basic input parameters. This lays the foundation for further verification, elimination of weakly correlated parameters, and identification of the basic input parameters. The specific names, units, and symbols of the average EGT of the No. 1 diesel engine cylinders and the other seven candidate basic input parameters are shown in Table 2.

Table 2 Parameter names, units, and symbols

Parameter Description	Unit	Symbol
Average cylinder exhaust gas temperature of diesel engine No. 1	℃	Tp
Exhaust outlet temperature of the turbocharger on diesel engine No. 1	℃	T1
Jacket cooling water outlet temperature of diesel engine No. 1	℃	T2
Low-temperature water air cooler outlet temperature of diesel engine No. 1	℃	T3
Lubricating oil inlet pressure of the turbocharger on diesel engine No. 1	bar	P1
Low-temperature water air cooler inlet pressure of diesel engine No. 1	bar	P2
Turbocharger speed of diesel engine No. 1	rpm	N
Prime mover load of diesel engine No. 1	%	L

Through the shipboard intelligent sensor system, approximately 200,000 raw data samples were collected. To analyze the correlation strength between the 7 candidate parameters collected from the actual "Xin Hong Zhuan" ship and the output feature diesel engine EGT, this study employs the Spearman's rank correlation method for analysis. This method is a non-parametric statistical tool that does not require the assumption of data normal distribution or linear relationships [43]. This characteristic gives it unique advantages when dealing with nonlinear data and datasets containing outliers [44], making it particularly suitable for the parameter relationship assessment needs of this study. This study aims to use this method to eliminate weakly correlated redundant parameters and retain core input parameters [45], ultimately reducing model parameter quantity and computational complexity while ensuring the explanatory power of features for EGT changes.

The Spearman correlation coefficient (denoted as r_s)is used to measure the strength and direction of the monotonic association between two variables, calculated as follows:

$$r_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{5}$$

where d_i represents the rank difference of the two variables for the i-th observation; n represents the total number of observations. The correlation coefficient r_s ranges from -1 to 1 [46]. A value closer to 1 or -1 indicates a stronger positive or negative monotonic correlation between the variables. To facilitate interpretation of the correlation analysis results, the study classified the association strength according to the standards given in Table 3. This classification framework provides support for systematically evaluating the interactions between variables in the dataset.

Table 3 Spearman correlation coefficient association strength classification standards

$\|r_s\|$ Range	Correlation Strength
0.8<$\|r_s\|$ <1	Very strong correlation
0.6<$\|r_s\|$ <0.8	Strong correlation
0.4<$\|r_s\|$<0.6	Moderate correlation
0.2<$\|r_s\|$ <0.4	Weak correlation
0<$\|r_s\|$ <0.2	Negligible correlation

Applying the Spearman method to the aforementioned dataset to explore the dependency relationships between diesel engine EGT and the 7 candidate basic input parameters, a correlation heatmap was generated as shown in Figure 3. In the heatmap, color intensity corresponds to the magnitude of the correlation coefficient: warmer colors (e.g., red) indicate stronger positive correlation, cooler colors (e.g., blue) indicate stronger negative correlation, and smaller correlations appear whiter.

Fig. 3 Spearman correlation coefficient heatmap for eight parameters

From the Spearman correlation coefficient heatmap in Figure 3, it can be seen that L, N, and T1 have significantly higher correlation strengths with the target variable Tp (diesel engine EGT) than the other candidate parameters. Their correlation coefficients with Tp 0.98, 0.98, and 0.96, respectively, all falling within the very strong correlation interval ($|r_s|$>0.8), providing data support for basic feature parameter screening.

From a physical mechanism perspective, these three parameters correspond to the core links of the diesel engine's combustion input, combustion conditions, and combustion verification. L determines the total heat release in the cylinder by adjusting fuel injection quantity [47], serving as the fundamental driving factor for EGT changes. N controls intake air density and oxygen supply, directly affecting combustion completeness [48]. T1, as a pre-feedback signal of combustion efficiency, can capture combustion abnormalities in advance [49]. These three jointly cover the key stages of the EGT generation process and are therefore selected as basic

input parameters. The parameter quantity is reduced by 60% compared to the initial candidate parameters, lowering model complexity while ensuring the features' explanatory power for EGT changes.

After screening the basic input parameters, approximately 30,000 time-series continuous actual ship data points were extracted from the collected raw data to form dataset E. The dataset was divided into training, validation, and test sets in a 7:1.5:1.5 ratio based on chronological order [50]. The selected data samples reflect various operational scenarios of the actual generating diesel engine, ensuring the general applicability of subsequent research. The training set is used to fit the model, the validation set helps adjust model hyperparameters, and the test set is used for independent verification of the final model's generalization ability and actual prediction performance. During training, the model's performance on the validation data is closely monitored, and the model configuration with the lowest prediction error and best match with actual values is selected.

The Interquartile Range (IQR) method is a non-parametric outlier detection technique [51] that does not rely on the assumption of normal data distribution, making it suitable for the skewed distributions and dynamic fluctuations commonly encountered in diesel-engine data. This method determines the normal data range by calculating the quartiles of the data, thereby identifying and handling outliers. During data cleaning, the application of the IQR rule effectively removes extreme noise while preserving the characteristics of real operational fluctuations.

Consequently, in terms of data cleaning, this study adopts a standardized process centered on eliminating invalid interference and retaining authentic operational conditions, tailored to the operating characteristics of marine diesel engines. First, invalid samples where data exceeded the physical operational boundaries of the actual diesel engine parameters were removed based on those boundaries, preliminarily retaining valid samples. Second, the IQR rule was used to target extreme values for each parameter within each interval [52]. Finally, for short-term, small amounts of missing data with continuous missing data points ≤5, the mean of the adjacent valid data points (one before and one after) was used for imputation. Only samples with continuous missing data points >5 were removed [53]. This data cleaning approach reduces the interference of extreme noise while preserving the continuity of time-series data and the fluctuation characteristics of normal operations, ensuring the model can learn the real operational characteristics of the diesel engine. It achieves a balance between ensuring modeling accuracy and processing efficiency and reasonableness.

3.2 LightGBM Model Design and Analysis

Based on the collected actual ship data, it was found that the diesel engine selected for this study operates within a stable load range of 0-80 %, with no operational scenarios exceeding this range. This load interval fully covers the entire actual operational state from economical speed and partial load to rated load. To accurately adapt to the real operational characteristics of the diesel engine and avoid interference from invalid load intervals in model training, this study focuses exclusively on the 0-80 % load range for modeling. Simultaneously, considering the significant heterogeneity in parameter correlation characteristics within different load intervals, to further improve model adaptation accuracy for each operating condition, the 0-80 % load interval is equally divided into 8 dedicated sub-intervals at 10 % intervals. Dedicated prediction models are constructed for each interval to ensure the models can accurately capture the variation patterns of EGT under different loads.

When training the 8 interval-specific LightGBM models, based on comparisons of the fit between actual predicted values and true values under different sliding window sizes and step lengths within the divided E1, E2, and E3 datasets, it was ultimately determined that the model training achieved the highest prediction accuracy when using a fixed chronological sliding window of 5 and a step length of 1.

Additionally, a three-level feature system is constructed. L, N, and T1 serve as basic features. Historical statistical features are introduced to enable the model to fully learn the dynamic fluctuation characteristics and stability of the core parameters. Parameter lag features are also added to adapt to the thermal inertia delay characteristic of diesel engine EGT. The three-level features collectively cover both static parameter correlations and dynamic temporal patterns, providing comprehensive and lightweight input support for the model. These features do not cause data leakage; the core reason is that feature construction is entirely based

on historically available data without introducing future information, ensuring the authenticity of model predictions. The specific input features for training the LightGBM model are shown in Table 4.

Table 4 Input features of the LightGBM model

Feature Tier	Feature Name	Symbol
Basic features	Diesel engine load	L
	Diesel engine turbocharger speed	N
	Diesel engine turbocharger exhaust outlet temperature	T1
Historical statistical features	Historical standard deviation of L within sliding window	L history std
	Historical standard deviation of N within sliding window	N history std
	Historical standard deviation of T1 within sliding window	T1 history std
Lag features	Load lagged by 1 step	L lag1
	Speed lagged by 1 step	N lag1
	Exhaust temperature lagged by 1 step	T1 lag1

Subsequently, Bayesian algorithm [54] is used for hyperparameter optimization of the dedicated models for different load intervals, customizing configurations according to the operational differences across the 8 load intervals. Core optimization parameters such as learning rate, maximum tree depth, and number of leaf nodes are iteratively searched to find the optimal mapping relationship between parameter combinations and model performance. Constraint conditions are set based on the data characteristics of each interval to determine the dedicated hyperparameters for each interval-specific model. L1 and L2 regularization are also introduced to suppress overfitting. Hyperparameter optimization provides foundational support for the continuous iteration of the model within the MLOps framework.

To verify that interval-specific LightGBM models can effectively improve prediction accuracy, this study conducted a performance comparison experiment between interval-specific and full-interval LightGBM models. To ensure a fair comparison, experimental conditions were controlled from three aspects: data, hyperparameters, and evaluation dimensions, while adapting the comparison experiment to the characteristics of interval-specific models.

Both types of models are trained and predicted based on the divided E1, E2, and E3 subsets of dataset E, ensuring a consistent comparison benchmark. For the full-interval case, data from E1 and E2 are first used to train and tune parameters to obtain the full-interval model. Then, the trained full-interval model is used to predict all samples in the entire test set, obtaining residuals for all data, and subsequently calculating the overall *MAE* and R^2 for the test set. For the interval-specific case, data from E1 and E2 are also used to train and tune parameters to obtain interval-specific models. The rule is: according to the 8 load interval division standards, each data row is sequentially placed into the corresponding interval for model training based on chronological order, resulting in eight interval-specific models. Then, data in E3 are assigned to the corresponding dedicated intervals based on their L values and chronological order, and the dedicated model for that interval is used for prediction to obtain corresponding residuals. Finally, the test set residual data from all intervals are aggregated to calculate the overall *MAE* and R^2. Through this method, the error metrics for both types of models are calculated based on the residuals of all test set samples, and the interval division standard for the interval-specific models is completely consistent with the evaluation scope of the full-interval model. Combined with visualization results, a fair comparison between the two is achieved under identical data coverage ranges and evaluation logic.

Hyperparameter configuration follows the principles of a unified optimization framework and personalized adaptation for each interval, ensuring fairness in the comparison between full-interval and interval-specific models. First, for the full-interval model, a reasonable optimization range for core hyperparameters is determined using the Bayesian optimization algorithm. This range covers the operational characteristics of different load intervals of the diesel engine, as detailed in Table 5. Subsequently, this

optimization range is used as the unified optimization benchmark for both the full-interval and interval-specific models. Both types of models independently perform Bayesian optimization within this range. The full-interval model finds the global optimal hyperparameters based on the entire dataset, while the 8 interval-specific models find the interval-specific optimal hyperparameters within the unified range, tailored to the operational characteristics of their respective load intervals.

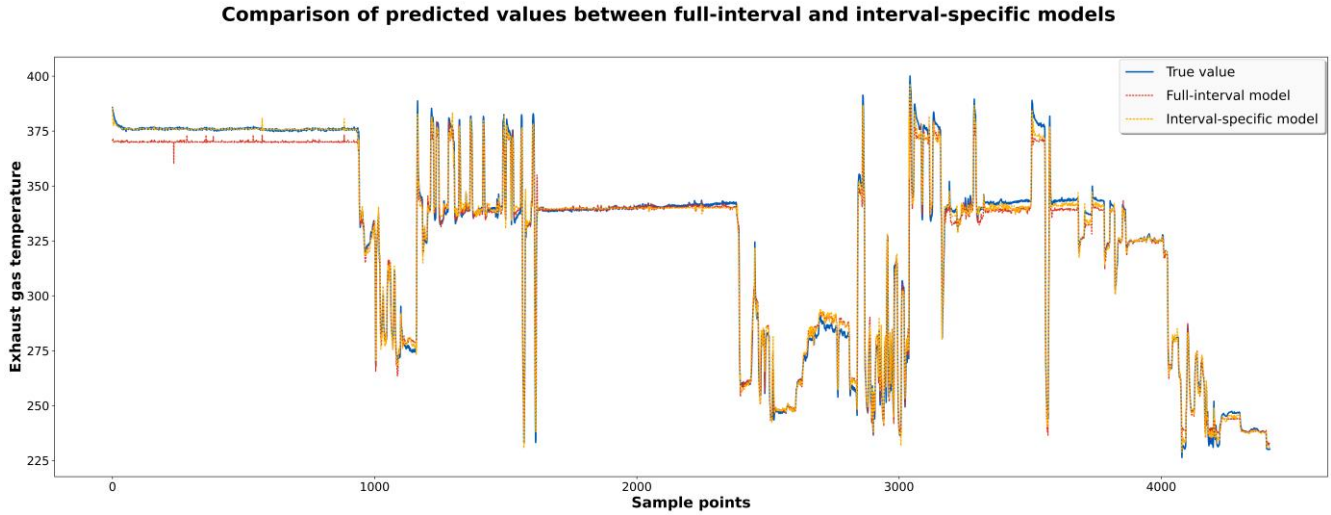
Table 5 Unified hyperparameter search space for the LightGBM model

Hyperparameter Category	Specific Parameter	Unified Search Range
Basic model configuration	Boosting_type	Gbdt
	Objective	Regression_l1
Core optimization parameters	Learning_rate	[0.01, 0.1]
	Max depth	[3, 10]
	Num leaves	[15, 130]
	Num boost round	[500, 2000]
Regularization parameters	Reg alpha	[0.1, 1.0]
	Reg lambda	[0, 1.0]
	Min split gain	[0.001, 0.1]
Sampling & Splitting parameters	Feature fraction	[0.6, 1.0]
	Bagging fraction	[0.7, 0.95]
	Bagging freq	[1, 7]
Leaf node control	Min child samples	[1, 20]
	Min child weight	[10, 25]
Other configurations	Max bin	[200, 300]
	Num threads	-1
	Seed	42
	Early stopping rounds	[20, 150]
	Sliding window size	5
	Prediction step	1

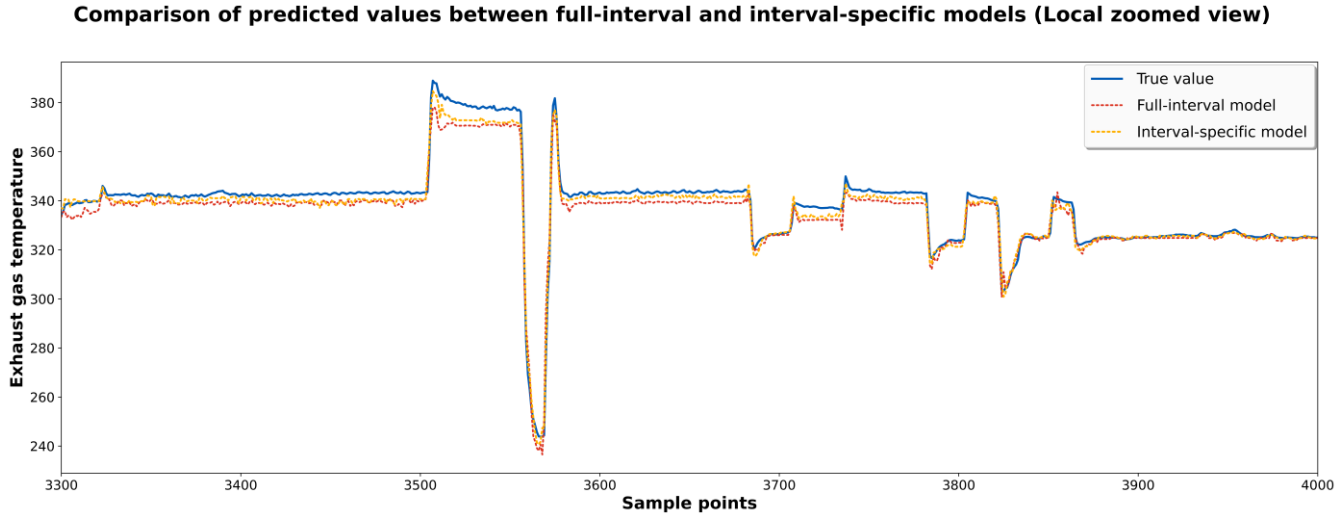
To further mitigate overfitting in the model training process and ensure the generalization ability of the interval-specific and full-interval models, a series of targeted control strategies based on validation set iteration and early stopping mechanism are strictly implemented in the Bayesian hyperparameter optimization process. In the training phase, the prediction error of the validation set is taken as the core index to monitor the model performance in real time, with the training of the corresponding hyperparameter combination terminated when the validation set error ceases to decrease, thus avoiding overfitting to the training set. A dedicated early stopping mechanism is also configured in the hyperparameter settings, where the training process is automatically stopped if the performance of the validation set does not improve for consecutive rounds within the range of 20~150 rounds, which prevents the model from learning the noise characteristics of the training set in the later training stage. Moreover, the Bayesian hyperparameter optimization in this study is not unrestricted search but customizes hyperparameters for each load interval within the unified search space that fits the operational characteristics of the diesel engine. This constrained optimization not only avoids overfitting caused by excessive hyperparameter tuning but also ensures the adaptability and structural simplicity of the model in each load interval.

This configuration method eliminates the interference of differences in basic parameter ranges on performance comparison, while ensuring that the trained interval-specific models can fully adapt to the data distribution characteristics of each load interval. Ultimately, a fair performance comparison between full-interval and interval-specific models under a unified optimization framework is achieved.

After completing the training of both full-interval and interval-specific models, the trained models are used to make predictions on test set E3. The final prediction performance comparison results are shown in Figure 4. To more clearly demonstrate the prediction accuracy of the full-interval and interval-specific models, the data points ranging from 3300 to 4000 in Figure 4(a) were selected to plot Figure 4(b).



(a) Comparison of predicted values between full-interval and interval-specific models



(b) Local zoomed view: Comparison of predicted values between full-interval and interval-specific models

Fig. 4 Prediction accuracy comparison between full-interval and interval-specific LightGBM models

The prediction comparison results in Figure 4 show that the interval-specific LightGBM model outperforms the full-interval LightGBM model on key metrics. The *MAE* decreases from 2.4348 to 1.8584, representing a reduction of 23.69 %. The R^2 improves from 0.9756 to 0.9866, an increase of 1.10 percentage points. Meanwhile, the *MSE* drops significantly from 14.7711 to 9.7828, corresponding to a 33.77 % decrease. Compared to the full-interval model's average fitting of heterogeneous operating conditions, the interval-specific model is custom-built according to the data distribution characteristics of each load interval, effectively reducing noise interference and feature underfitting across different load intervals. This confirms the rationality and engineering applicability of the interval-specific LightGBM model.

To further verify the comprehensive performance of the interval-specific LightGBM model in prediction accuracy, real-time inference and engineering deployment, this study conducts a horizontal comparison with typical single deep learning models applied in the frontier field of marine diesel engine EGT prediction, with the specific comparison indicators shown in Table 6.

Table 6 Performance comparison of different models for EGT prediction

Model	*MAE*	Inference Delay (ms)	Docker Image Size (MB)
LightGBM	1.8584	1.026	480
CNN	1.7257	78.389	1250
LSTM	1.6643	76.198	1140
BiGRU	1.5856	155.335	1700
BiLSTM	1.5328	70.969	1060

All metrics in Table 6 are tested in a simulated marine edge server environment under unified resource constraints; the inference delay refers to the time consumption of real-time inference for a single sample, and the Docker image size represents the final volume of the model after lightweight containerization.

Test results show that although the LightGBM model, as a machine learning method, has a slightly higher *MAE* value and thus marginally lower prediction accuracy compared with the involved deep learning models, it exhibits an overwhelming advantage in real-time inference performance and lightweight engineering deployment. Specifically, its single-sample inference delay is only 1.026 ms, which is reduced by approximately 98.56 % compared with BiLSTM, the optimal deep learning model with an inference delay of 70.969 ms, fully meeting the millisecond-level real-time monitoring requirements for EGT prediction of marine diesel engines in intelligent engine rooms. In terms of the Docker image size after lightweight containerization, the LightGBM model is merely 480 MB, less than half of the 1060 MB of BiLSTM—the smallest deep learning model—and accounts for only about 28.2 % of BiGRU, the largest one. This ultra-lightweight volume not only greatly reduces the storage resource occupation of shipboard edge servers with limited hardware resources, but also significantly shortens the model deployment and iteration time in the local automated pipeline, effectively enhancing the engineering implementability of the model in the network-constrained marine edge environment. Therefore, LightGBM, as a machine learning model, demonstrates remarkable comprehensive advantages in the scenario of engineering deployment.

3.3 Dynamic Threshold Design

To improve the anomaly detection accuracy of the warning system and reduce the risk of false alarms and missed detections, this study designs two types of thresholds for different load intervals: the dynamic threshold for EGT residuals (Thr1) and the threshold for the rate of change of core input parameters (Thr2).

To determine Thr1, this paper proposes a dynamic threshold setting method that integrates the Interquartile Range (IQR) and an interval-specific adjustment coefficient α. This method is based on the classic Tukey's Fences method [55], whose threshold boundary is defined by the following equation:

$$\text{Threshold} = [Q_1 - k \times IQR,\ Q_3 + k \times IQR] \tag{6}$$

where Q1 and Q3 are the first and third quartiles of the data, respectively, IQR = Q3 - Q1 is the interquartile range, and k is the Tukey multiplier, whose standard empirical value is k = 1.5.

First, based on the residuals predicted by the interval-specific LightGBM model on dataset E, Q1, Q3, and IQR of the residuals for each load interval are calculated. The basic threshold range (Q1−1.5×IQR, Q3+1.5×IQR) is constructed according to Equation (6).

However, as shown in Figure 5, the residual distributions of different load intervals exhibit significant heterogeneity. This indicates that in complex multi-condition systems, the global assumption of a fixed multiplier k=1.5 is difficult to achieve optimal detection for all load conditions [56]. To solve this problem, this paper innovatively introduces an adjustment coefficient α related to the specific load characteristics of each interval, customizing the scaling of the standard Tukey multiplier, thereby forming a residual dynamic threshold formula adapted to the characteristics of each interval:

$$Thr1 = [\,Q1 - \alpha \times 1.5 \times IQR, Q3 + \alpha \times 1.5 \times IQR\,] \tag{7}$$

In Equation (7), the equivalent dynamic multiplier (α×1.5) replaces the fixed multiplier k in Equation (6). The coefficient α is set to quantify the operational differences between different load intervals (such as load stability and the distribution characteristics of model prediction errors), enabling fine-tuning on the basic threshold. Ultimately, by calculating and applying their respective dedicated α values for the eight load intervals, Thr1, which is highly adapted to each interval, is generated, significantly improving the adaptability and reliability of the anomaly detection system to the complex and variable operating conditions of the diesel engine.

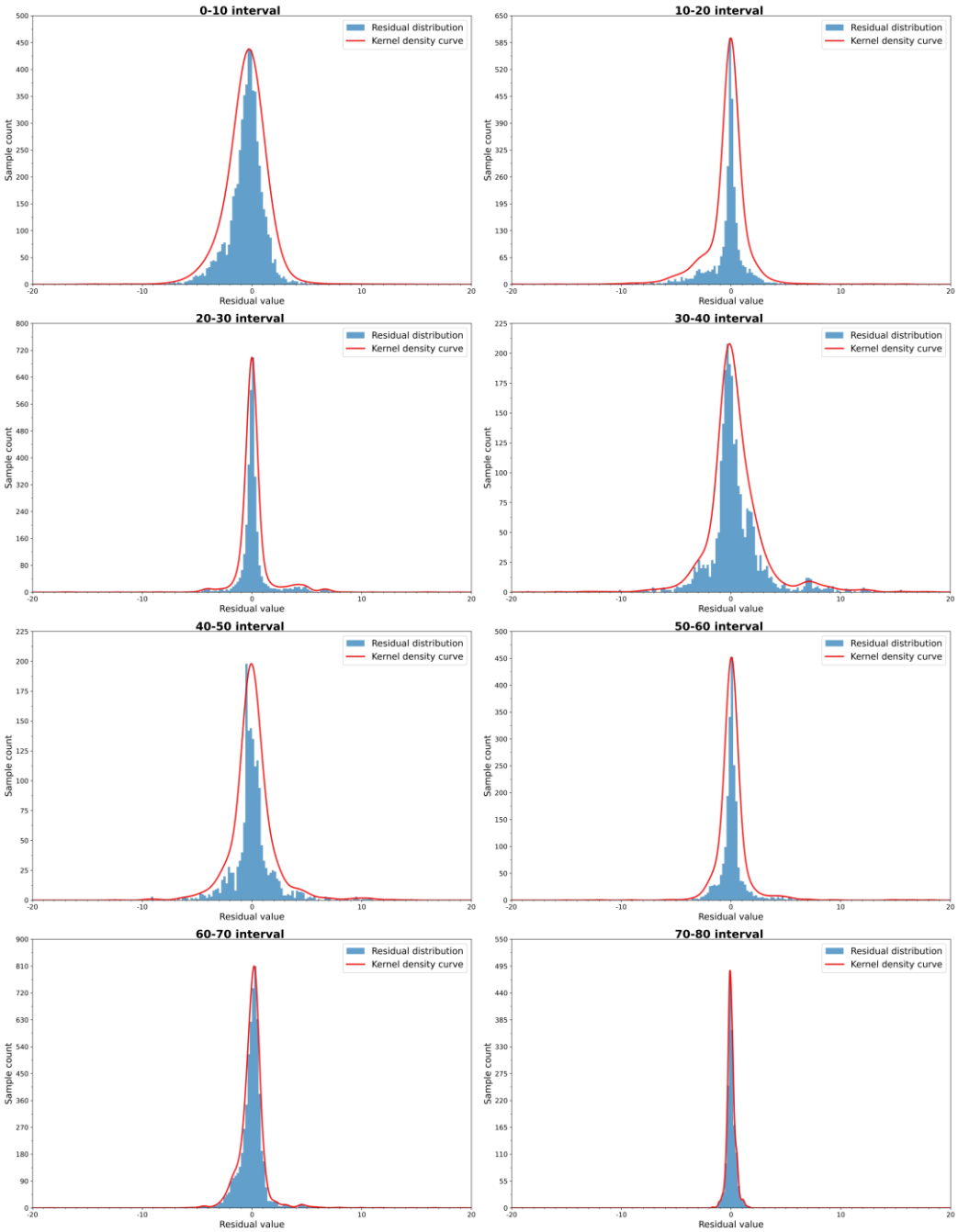


Fig. 5 Residual Distribution of Diesel Engine Exhaust Gas Temperature across Eight Intervals

In the design of Thr1, the normal condition coverage rate (Coverage Rate) is the core indicator for measuring the effectiveness of the dynamic threshold range. It is defined as the percentage of residuals from diesel engine data under normal operating conditions falling within the preset threshold range for a given load interval. As shown in Figure 5, there are significant differences in the dispersion of residuals across different

load intervals. Therefore, the target coverage rate needs to be adapted to the characteristics of each interval [57] to balance anomaly detection accuracy with the risk of false and missed alarms. Specifically: If the target coverage rate is set too high, for intervals with greater residual dispersion, overly wide thresholds can easily misclassify abnormal states as normal fluctuations, leading to missed alarms; for intervals with gentle residual fluctuations, overly wide thresholds may fail to effectively identify subtle abnormal residuals, also increasing the risk of missed alarms. Conversely, if the target coverage rate is set too low, for intervals with greater residual dispersion, overly narrow thresholds will significantly increase the probability of normal data being falsely reported as abnormal; for intervals with gentle residual fluctuations, overly narrow thresholds will misclassify normal fluctuations as abnormal, thereby compromising the stability of the warning system.

Therefore, to balance false and missed alarms, with the overall coverage rate controlled between 90% and 99% as a constraint, multiple sets of comparative experiments were conducted to optimize α. Combined with comprehensive indicators of false alarm rate and anomaly detection rate for each interval, the optimal α parameters for each interval were finally determined, as shown in Table 7.

Table 7 Optimized interval-specific adjustment coefficients (α) and corresponding normal data coverage rates

Load Interval	Interval-Specific Adjustment Coefficient α	Normal Data Coverage Rate (%)
0-10	1.3	95.20
10-20	1.9	92.42
20-30	1.8	92.88
30-40	1.7	94.68
40-50	1.8	93.79
50-60	1.9	92.51
60-70	1.5	95.68
70-80	1.2	96.08

Additionally, Thr2 is also designed dynamically, incorporating the rate of change of L, N, and T1. Within the daily iteration process of MLOps thresholds, the benchmark threshold for the rate of change in each interval is anchored based on the previous day's actual ship normal operating data. Window period calibration adapts to parameter fluctuation characteristics, avoiding the defect of traditional fixed thresholds being unable to respond to dynamic changes in ship operating conditions, providing precise support for rate-of-change anomaly detection.

Its core logic is to preset a rate-of-change benchmark threshold adapted to the operating conditions of each load interval based on the statistical characteristics of historical normal data for each interval. This benchmark value is determined daily by calculating the 95th percentile range of parameter change rates within the corresponding load interval from the previous day's actual ship normal operating data. Specifically, the 5th percentile and the 95th percentile are taken as the lower and upper limits of normal fluctuation, precisely fitting the interval distribution characteristics of industrial data. Simultaneously, a window period calibration mechanism is introduced. During daily iteration, the degree of parameter fluctuation dispersion within the time-series sliding window is calculated. If the dispersion exceeds 1.2 times the interval mean, the adjustment range is expanded by "benchmark threshold interval ×1.2" to filter instantaneous noise; if it falls below 0.8 times the interval mean, the adjustment range is narrowed by "benchmark threshold interval ×0.8" to improve anomaly capture efficiency. Ultimately, through the synergy of daily iteratively updated benchmark thresholds and window period adjustments, Thr2 achieves precise adaptation to parameter fluctuations during the dynamic operation of the ship, fully conforming to the complex distribution characteristics of industrial data.

3.4 Warning Logic Design

The anomaly detection of the warning system is designed using a three-level warning logic [58]. Based on the heterogeneity of parameter correlations across different load intervals, anomaly detection employs differentially designed warning sliding windows: low-load intervals use a 20-step sliding window, medium-

load intervals use a 15-step sliding window, and high-load intervals use a 10-step sliding window. This interval-specific configuration enhances noise tolerance and the ability to capture subtle anomalies. To capture the dynamic fault characteristics that are difficult to fully reflect in the absolute value of predictions, the system incorporates the rate of change of the three core parameters as independent diagnostic indicators. These are denoted as L rate (load change rate), N rate (turbocharger speed change rate), and T1 rate (turbocharger exhaust temperature change rate). Through a three-level warning system based on absolute value anomalies, rate-of-change anomalies, and three-parameter coupled anomalies, and by setting warning buffer zones at interval boundaries, a balance between warning sensitivity and stability under different loads is achieved [59], enabling precise warning.

(1) Level 1 Warning: Slight Fluctuation Alert

Invoking the interval-specific residual dynamic threshold from Section 3.3, when the residual exceeds the corresponding interval threshold for 2 consecutive time steps, a Level 1 warning is marked, and the anomaly direction is recorded, preliminarily screening potential anomaly samples.

(2) Level 2 Warning: Abnormal Trend Warning

Based on the warning sliding window mechanism, trend verification is performed on Level 1 warning samples. When the proportion of valid Level 1 warning samples within the window is ≥40%, and the anomaly direction remains consistent, rate-of-change collaborative verification is initiated. For the three-core parameter change rates—L rate, N rate, and T1 rate—interval-specific IQR threshold verification is used. When the proportion of samples where any parameter change rate exceeds the threshold is ≥20% and the direction is consistent with the residual anomaly direction, it is marked as a Level 2 warning.

(3) Level 3 Warning: Fault Precursor-Level Alarm

To adapt to the complex distribution characteristics of industrial data, which are often noisy and exhibit heterogeneous fluctuations, the residual threshold for Level 3 warning is set to 1.2 times the Level 1 warning residual threshold. For samples that have already triggered a Level 2 warning, two conditions must be met simultaneously to be judged as a Level 3 warning. First, the proportion of Level 2 warning samples within the warning sliding window must reach 70 % or higher. Second, the current residual must exceed the preset Level 3 warning residual threshold. Simultaneously, the system associates core parameter change rate features. The flowchart of the three-level warning system is shown in Figure 6.

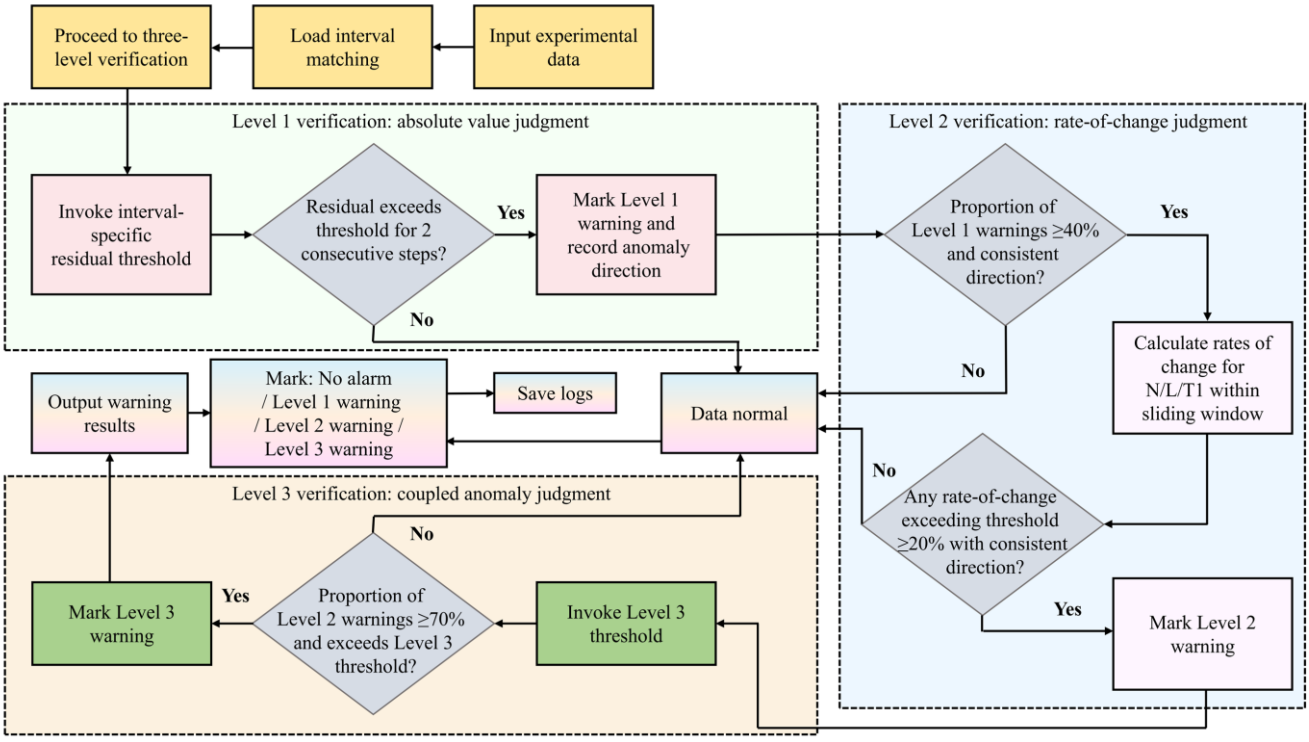


Fig. 6 Flowchart of the three-level warning system

Before executing the dual-condition judgment for the third-level warning, based on the load value of the current sample, it is strictly matched to the corresponding interval and assigned a unique label, prioritizing adherence to the interval boundary buffer zone rule. If the load falls within the ±1% buffer zone of an interval boundary, the interval label from the previous sample is directly reused to avoid frequent cross-interval label switches caused by minor load fluctuations. When traversing samples in chronological order, only when the load exceeds the ±1 % buffer zone range, causing the current sample's interval label to differ from the previous one, is that continuous cross-interval segment marked. This segment temporarily does not participate in third-level warning judgment and only triggers a first-level warning. After subsequent samples remain stably within the same interval for 2 consecutive time steps, the third-level warning judgment process is restarted. Ultimately, only sample segments with continuously consistent interval labels are retained for third-level warning judgment. Moreover, the calculation of core parameter change rates uses only parameter differences between consecutive samples within the same interval, completely eliminating cross-interval data interference and ensuring the accuracy of change rate calculation and warning judgment.

3.5 Warning and Fault Identification Effectiveness Analysis

To verify the accuracy and practicality of the three-level warning system designed in this study, effectiveness validation was conducted based on dataset E. Figure 7 shows two segments of normal data residuals extracted from E. Within these two residual segments, two sub-segments were selected, and simulated anomaly data modeled on real-world failure mechanisms were introduced into each, simulating two types of fault precursors: mild turbocharger blade fouling and fuel injector faults [60], to verify the sensitivity and warning capability of the warning system. Specifically, the simulated fuel injector fault in this study refers to the common condition where the fuel injector becomes stuck in the closed position due to carbon deposition. Figure 8 shows the two residual segments after introducing the two types of faults. These two types of faults are typical and frequently occurring failure modes in the actual operation of marine diesel engines, and their early fault characteristics are highly correlated with abnormal changes in EGT, which are thus selected as the research objects for simulation verification.

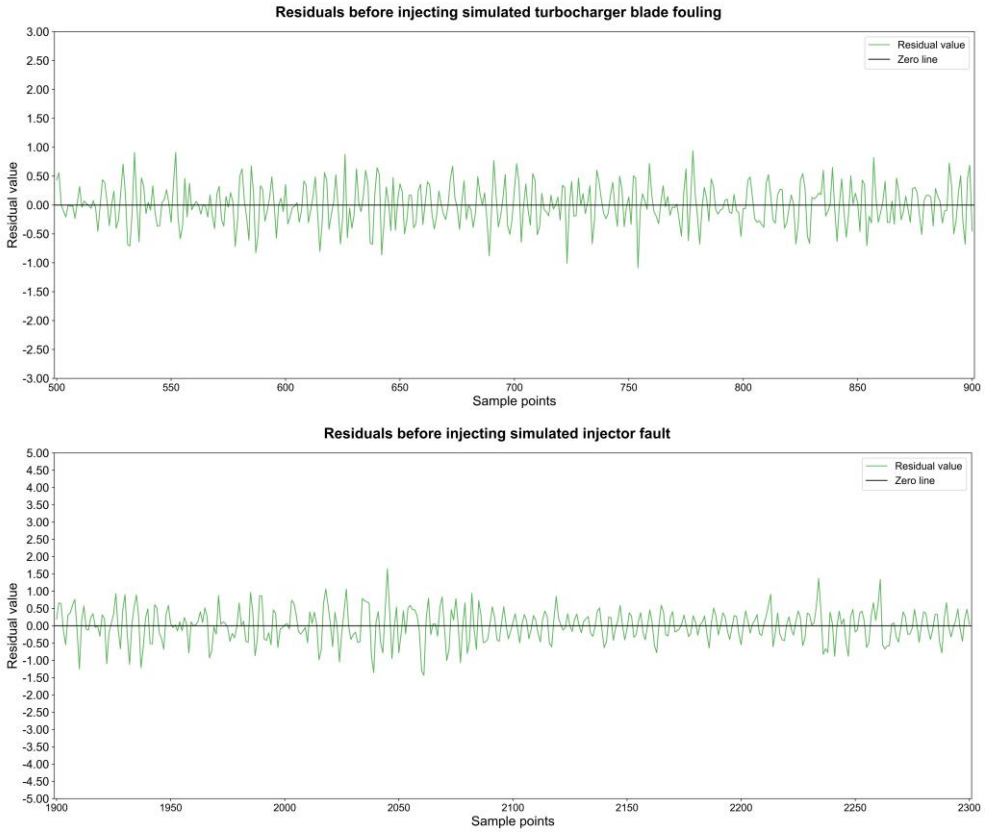


Fig. 7 Residuals of Diesel Engine Exhaust Gas Temperature under Normal Operating Conditions

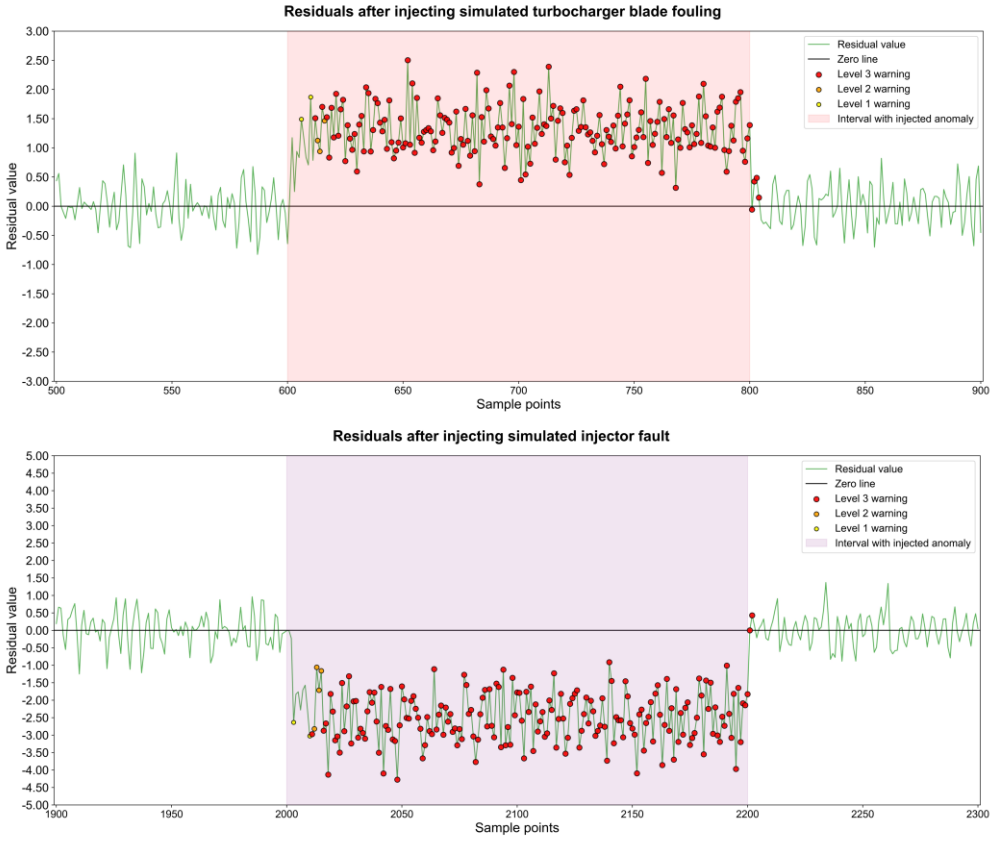


Fig. 8 Residuals after Injecting Simulated Faults and the Corresponding Warning Levels

As can be seen from Figure 8 after introducing anomalies, the three-level warning system can effectively identify the anomalous data. The recognition rates for the two simulated anomalies—mild turbocharger blade fouling and fuel injector faults—are 92.04 and 94.53 %, respectively. Simultaneously, the three-level warning system produces very few false alarms for the normal data before and after the two sub-segments where anomalies were introduced, and false alarms exist only at the edges between the normal data intervals and the simulated anomaly data intervals. This fully demonstrates that the warning system can accurately capture early fault precursors of the diesel engine while effectively controlling false alarm risks, possessing both high sensitivity and strong stability, fully adapting to the equipment health monitoring needs under complex real-ship operating conditions.

4. MLOps Application

4.1 MLOps Design

Based on the general three-stage MLOps principle described in Section 2.2, to adapt it to shipboard offline, low-computing-power edge scenarios and achieve engineering integration with the warning system, this section proposes a four-layer MLOps system architecture. This architecture is a concrete implementation and scenario-specific extension of the three-stage principle. The environment layer provides foundational support for the entire process. The data layer and iteration layer jointly undertake and refine the core tasks of the ML stage and integrate the continuous integration concept of the Dev stage. The operation layer fully corresponds to the responsibilities of the Ops stage.

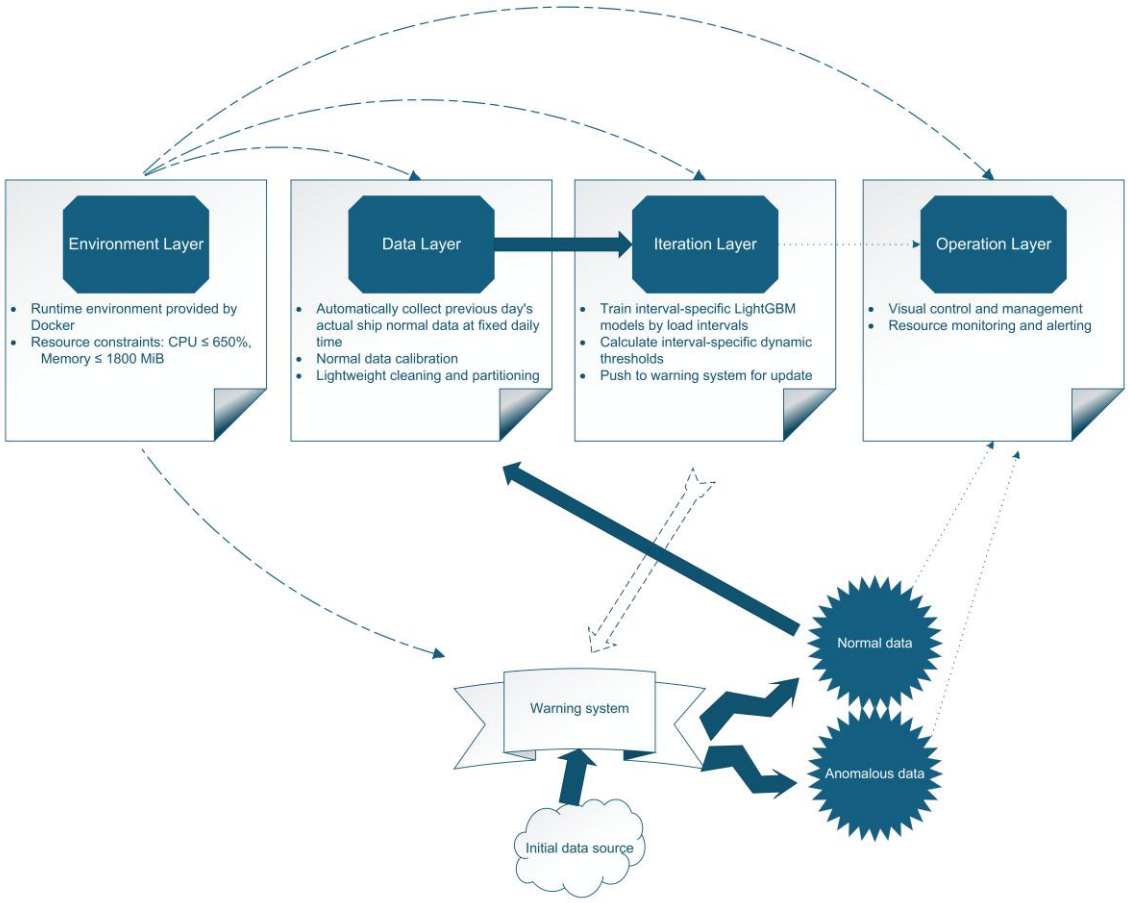


Fig. 9 Four-Layer MLOps System Architecture for the Warning System

As shown in Figure 9, the four-layer architecture comprises distinct functional layers. The environment layer provides a standardized, low-resource operating environment for all layers and the warning system through Docker multi-stage builds and resource constraints, avoiding environment compatibility issues. The data layer automatically collects the previous day's actual ship normal data (data for which the warning system did not trigger a third-level warning) at a fixed time daily. After calibration and the cleaning process described in Section 3.1, the data is divided into 8 data subsets according to the 10% load intervals, versioned, and output as high-quality data to the iteration layer. Simultaneously, anomalous data is archived for subsequent retrospective analysis. The iteration layer takes the data layer's output as input, trains dedicated LightGBM models for the 8 intervals, generates dynamic thresholds and warning parameters for each interval, and finally stores and pushes the new models, new thresholds, and warning parameters bound together to the warning system. It allows one-click rollback to the previous latest normal model and threshold when the newly trained model and thresholds are abnormal. The operation layer enables visual control of the entire process through an Excel configuration table, allowing adjustments to the data layer cleaning rules and configuration of iteration layer hyperparameters, while monitoring resource usage and triggering local alerts when approaching thresholds.

To promote the engineering implementation and long-term operation of the marine diesel engine EGT warning system, an end-to-end automated iteration mechanism is constructed with MLOps as the core framework. The core goal is to dynamically update warning models and anomaly detection thresholds through this mechanism, adapting to dynamic scenarios such as equipment aging and gradual operational drift, while ensuring the consistency and reliability of real-time warning system judgments. This provides comprehensive technical support for fault prediction in ship intelligent engine rooms. To ensure the design can pass engineering verification and align with real-ship scenarios, this design uses an ordinary office computer

equipped with an Intel Core i7-10875H CPU and 16GB RAM as the core carrier to simulate a real-ship edge server. Around this simulated carrier and the operational characteristics of real ships, three core constraints are designed to achieve synergistic adaptation between the iteration process and the warning system.

First, we present the hardware resource adaptation design. Addressing the prevalent low computing power characteristics of ship edge servers [61], the aforementioned ordinary office computer is used as the standard carrier for simulating the real-ship environment, ensuring that the computing power level of the simulated carrier matches that of the real-ship edge server. Through software-level resource scheduling optimization, the total resource usage of all MLOps layers and the warning system is strictly limited.

Second, we propose the offline closed-loop operation design. To adapt to the scenario characteristics of some ships operating without external networks, only within a closed local area network during navigation, a local independent closed-loop process is constructed with the simulated carrier as the core. The data acquisition module imports actual ship data through simulated interfaces to simulate the effect of local real-time sensor data acquisition. The model training and threshold calculation modules complete iterations based on historical data stored locally on the simulated carrier and real-time imported data [62]. Warning system parameter updates are achieved through local process communication on the simulated carrier. The entire process does not rely on cloud services or real-time streaming tools, ensuring from the design level that the system can still perform anomaly detection functions normally under simulated network-disconnected scenarios, fully aligning with the requirements of real-ship closed operation.

Finally, we introduce the equipment aging adaptation design. Using the simulated carrier as the operational foundation, the core function of the MLOps iteration mechanism is focused on dynamic adaptation capability. Supporting validation logic is designed to ensure adaptation effectiveness. A short cycle is used to simulate the complete daily iteration process of a real ship [63], while an aging simulation scenario is constructed based on the imported actual ship data [64]. An equipment aging simulation experiment is designed with two control groups. The experimental group enables the daily iteration mechanism, retraining dedicated LightGBM models for the 8 load intervals and updating dynamic thresholds based on previous normal data. The control group uses only the initial model and fixed thresholds, without performing any iteration updates.

4.2 MLOps Application Verification

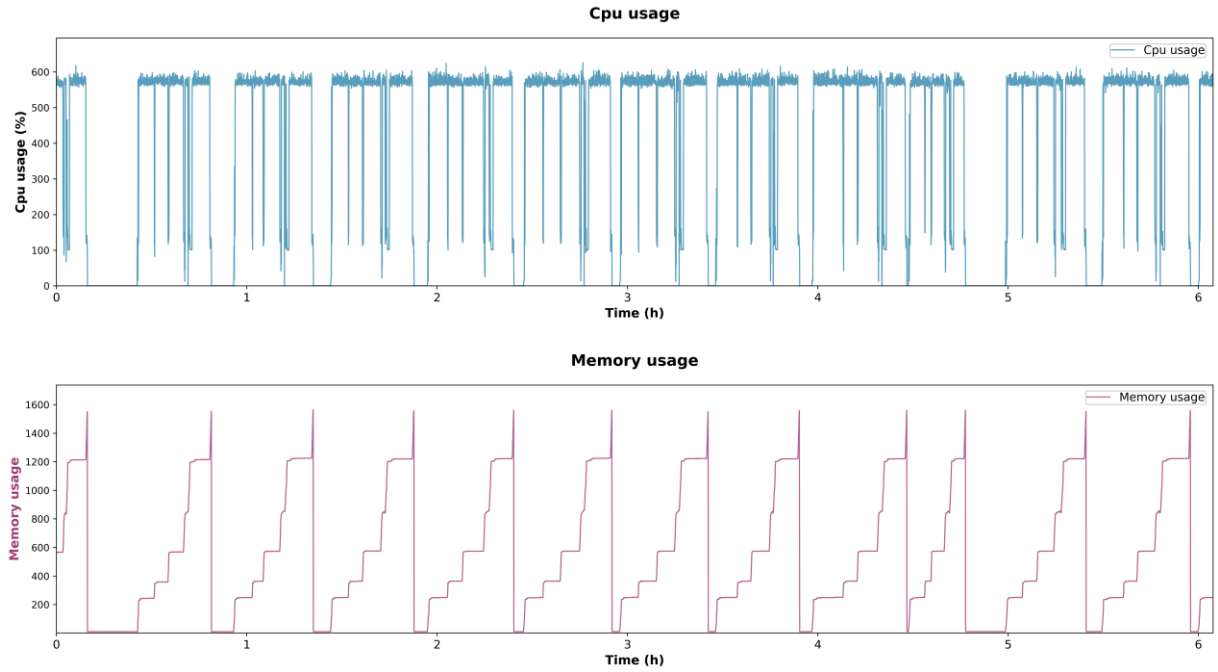
To comprehensively verify the feasibility of the solution's engineering implementation, this study uses the complete dataset E as input and conducts verification from three core dimensions—offline closed-loop operation, hardware resource adaptation, and equipment aging adaptation—on the aforementioned simulated carrier. Simultaneously, resource constraints are applied to each layer, with specific constraint parameters as shown in Table 8. Verification for each dimension is designed with dedicated processes tailored to real-ship operational characteristics.

Table 8 Resource constraints for MLOps layers and the warning system

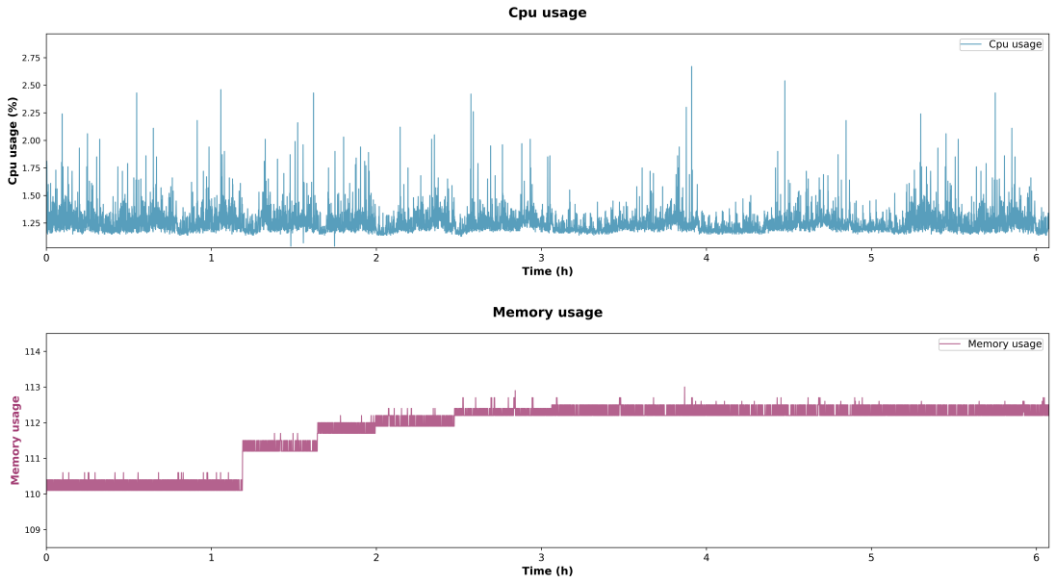
Component	CPU Limit (%)	Memory Limit (MiB)
Data Layer	⩽30	⩽50
Iteration Layer	⩽600	⩽1600
Operation Layer	⩽10	⩽50
Environment Layer	⩽5	⩽50
Warning System (Real-time prediction)	⩽5	⩽150

First, the local automated pipeline mode of the warning system under MLOps is activated. The data layer and iteration layer are set to execute approximately every 30 minutes. The warning system extracts one data row from E every second for prediction to simulate the uninterrupted prediction of the real-ship warning system, simulating the scenario of daily scheduled updates and real-time warning of the warning system on

the ship. The resource usage of all MLOps layers and the real-time warning system is monitored, with data points recorded at three-second intervals. All network connections are disconnected during the fully automated pipeline run to simulate the real-ship offline environment. Since the iteration layer in MLOps and the real-time update process of the warning system are most prone to stuttering during operation, observing the operation of these two layers can determine whether the entire process experiences stuttering. Figure 10 shows the resource usage of the iteration layer and the real-time warning system update captured over six hours.



(a) Iteration layer resource usage



(b) Resource usage of the warning system during real-time prediction

Fig. 10 Resource usage during six-hour offline operation

As seen from Figure 10, there is no stuttering during iteration updates in the iteration layer or during real-time updates in the warning system, only fluctuations within the normal range. This verifies stability during offline closed-loop operation.

Subsequently, the resource usage of all MLOps layers and the real-time warning system update during the six-hour automated pipeline run is statistically aggregated and merged, as shown in Figure 11.

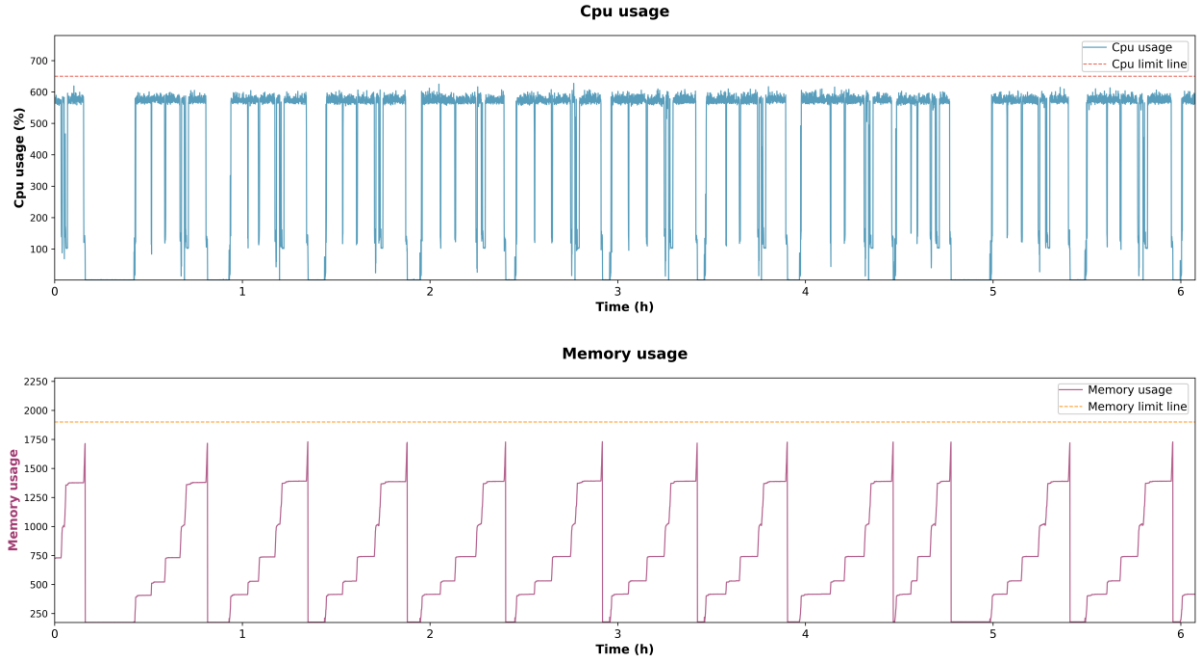


Fig. 11 Aggregated resource usage of the complete MLOps pipeline

As seen from Figure 11, the entire automated pipeline of the warning system under MLOps has CPU usage less than 650 % and memory usage less than 1900 MiB, both not exceeding the preset resource limits, verifying hardware resource adaptation.

To verify whether the MLOps daily iteration update mechanism can offset the parameter drift impact caused by ship equipment aging through dynamic model optimization, a 180-day model training experiment was conducted based on dataset E to simulate the natural aging process of equipment over 180 days. The 180-day duration is selected as it fully covers the typical early-to-gradual aging stage of ship equipment and enables sufficient verification of the long-term anti-drift capability of the MLOps iterative mechanism under the constrained hardware resources of shipboard edge environments. The experiment used code to construct an aging model integrating the logic of basic noise exponential decay growth and fluctuating noise dynamic adjustment, accurately replicating aging characteristics such as turbocharger wear and sensor drift in real-ship scenarios. Noise was introduced into the system after the first day's training was completed, and the noise intensity was slightly increased daily based on the previous day's level to simulate the continuous deepening of the aging degree. The experiment set up two control groups. The experimental group enabled the daily iteration mechanism, retraining dedicated LightGBM models for the 8 load intervals based on the normal operating data from the previous 10 days. The control group consistently used the initially trained model without performing any iteration updates. The effectiveness of the iteration mechanism was ultimately verified by comparing the *MAE* trend changes between the two groups.

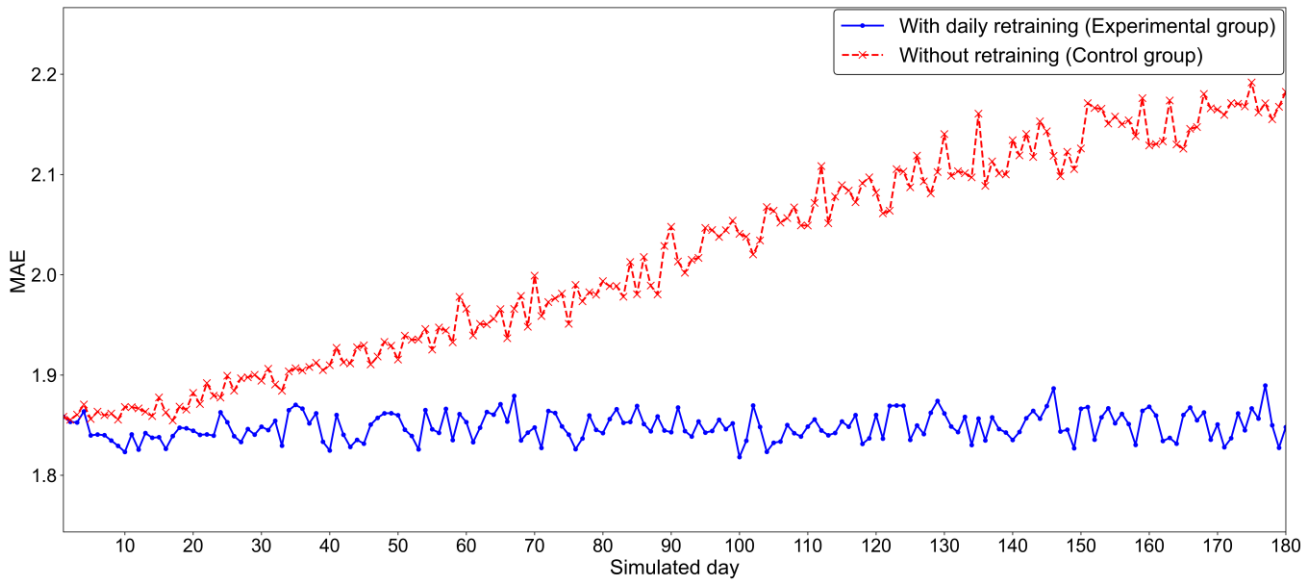


Fig. 12 *MAE* Trend Comparison with vs. without Daily Retraining under Simulated Equipment Aging

The experimental results in Figure 12 show that the *MAE* of the experimental group remained at a low fluctuation level throughout the 180-day cycle, with prediction error not showing significant degradation due to equipment aging, and overall accuracy remained stable. In contrast, the *MAE* of the control group showed a significant upward trend with the aging process. If operated long-term, its prediction residuals would become increasingly large, leading to ineffective identification of equipment anomalies. In conclusion, the MLOps daily iteration mechanism can effectively resist parameter drift problems caused by ship equipment aging by relying on a data-driven continuous model update strategy, ensuring the accuracy stability of the warning system throughout the equipment's entire lifecycle, significantly improving prediction accuracy, fully adapting to the engineering requirements of long-term real-ship operation, and verifying the strong resistance capability of this iteration mechanism to ship equipment aging.

5. Conclusion

This study addresses three key unresolved technical pain points in the research on EGT warning systems for marine diesel engines, and proposes a comprehensive solution. The core research results of the solution are as follows:

(1) Innovatively constructed interval-specific LightGBM prediction models achieved a test set *MAE* of 1.8584, *MSE* of 9.7828 and R^2 of 0.9866; compared with the full-interval model, the *MAE* was reduced by 23.69 %, *MSE* by 33.77 % and R^2 increased by 1.10 percentage points, significantly enhancing EGT prediction accuracy under variable load conditions. This performance gain arises from the models' ability to adapt to heterogeneous data distributions across load intervals, enabling precise capture of EGT variation patterns.

(2) The lightweight LightGBM model demonstrates superior engineering comprehensive performance, with a single-sample inference delay of only 1.026 ms that fully meets the millisecond-level real-time monitoring requirements for marine diesel engine EGT prediction in intelligent engine rooms. This inference speed is far superior to that of mainstream deep learning models for EGT prediction, presenting an overwhelming advantage in real-time response. Its ultra-low latency and compact structure further facilitate seamless integration with on-board real-time monitoring systems.

(3) The designed three-level logic dynamic threshold warning system achieves a 92.04 % fault identification rate for mild turbocharger blade fouling and a 94.53 % rate for fuel injector faults in simulated tests. It effectively balances the sensitivity and stability of anomaly detection while controlling the risk of false alarms. Its interval-adaptive threshold adjustment and multi-dimensional verification logic ensure reliable performance during complex operating transitions.

(4) Tailored to shipboard edge scenarios, the four-layer MLOps architecture containerizes the model into a 480 MB lightweight image, maintaining CPU usage below 650 % and memory usage below 1900 MiB. A 180-day simulated aging test confirmed its resistance to parameter drift caused by equipment aging. It sustains long-term stable prediction accuracy and enables automated model iteration and deployment in network-constrained environments.

This study effectively solves the problem of insufficient adaptability of a single model to variable load conditions through interval-specific modeling. It also compensates for the defect that fixed thresholds are difficult to adapt to parameter differences under variable loads through dynamic threshold design. Meanwhile, it achieves the engineering deployment of the warning system in shipboard edge environments through lightweight modeling and MLOps architecture, meeting the real-time monitoring needs of actual ships. It provides a feasible technical scheme for equipment health management and early fault warning in real-ship intelligent engine rooms and also offers new insights for the digital and intelligent health management of key equipment in marine intelligent engine rooms.

Building on the verified technical merits of this study, we propose actionable practical management and application recommendations for key maritime stakeholders, aiming to translate research outcomes into practical operational value. Marine equipment developers can refer to the research outcomes of this study such as interval-specific modeling, lightweight containerized design and three-level warning logic to improve their marine monitoring products. If the warning system of this study is ultimately deployed on actual ships, marine engineers can leverage the system's pre-fault signals to achieve early fault warning. They can integrate the warning prompts with routine inspection workflows. They can prioritize checks on high-risk components when alerts are triggered, thereby enhancing the efficiency and accuracy of fault diagnosis and troubleshooting in the engine room.

At present, this study still has certain limitations. The load intervals are merely divided by fixed equal spacing for interval-specific modelling, failing to further refine the intervals in accordance with the non-uniform variation characteristics of different load stages. In addition, the entire warning system and MLOps architecture have only been verified via simulated deployment based on the real-ship hardware environment with offline collected operational data, without the completion of actual on-board deployment and long-term online operation verification under real navigation conditions. To address the above shortcomings, subsequent research plans will focus on two key directions. On the one hand, clustering algorithms will be introduced to conduct data-driven adaptive division of diesel engine load intervals, replacing the current fixed equal-spacing division method to make the interval-specific models more consistent with the actual operational characteristics of diesel engines. On the other hand, the developed intelligent EGT warning system will be officially deployed on actual ships for long-term online operation testing, and the model parameters, dynamic thresholds and warning logic will be continuously optimized based on real-time collected ship operational data and actual fault information, so as to verify the engineering applicability of the system in real maritime navigation scenarios. Efforts will be made to conduct more in-depth research on the health management of marine diesel engines in intelligent engine rooms.

ACKNOWLEDGMENTS

This research was supported by the National Key R&D Program of China (grant number 2022YFB4301400), and the High-technology Ship Research Program (grant number CBG3N21-3-3).

REFERENCES

[1] Hoang, A. T., Chen, W. H., Lopez-Escalante, M. C., Guerrero-Pérez, M. O., Rodríguez-Castellón, E., Kowalski, J., Le, T. T., Bui, V. G., Nguyen, X. P., 2026. Methanol for decarbonization of the maritime sector: From ideological strategy to practical solutions. *Renewable and Sustainable Energy Reviews*, 229, 116638. https://doi.org/10.1016/j.rser.2025.116638

[2] Tuswan, T., Sari, D. P., Muttaqie, T., Prabowo, A. R., Soetardjo, M., Murwantono, T. T. P., Utina, R., Yuniati, Y., 2023. Representative application of LNG-fuelled ships: a critical overview on potential GHG emission reductions and economic benefits. *Brodogradnja*, 74(1), 63-83. https://doi.org/10.21278/brod74104

[3] Lu, Z., Zeng, H., Zheng, Z., Li, C., Dong, X., 2025. Research on deep learning-based network intrusion detection methods for smart ships. *Brodogradnja*, 77(2), 77205. https://doi.org/10.21278/brod77205

[4] Jeon, J., Theotokatos, G., 2024. A framework to assure the trustworthiness of physical model-based digital twins for marine engines. *Journal of Marine Science and Engineering*, 12(4), 595. https://doi.org/10.3390/jmse12040595

[5] Hoang, A. T., Pandey, A., Martinez De Osés, F. J., Chen, W. H., Said, Z., Ng, K. H., Agbulut, Ü., Tarełko, W., Olçer, A. I., Nguyen, X. P., 2023. Technological solutions for boosting hydrogen role in decarbonization strategies and net-zero goals of world shipping: Challenges and perspectives. *Renewable and Sustainable Energy Reviews*, 188, 113790. https://doi.org/10.1016/j.rser.2023.113790

[6] Guan, W., Gu, J. K., Pan, X. B., Pan, M. Z., Wang, X. Y., Zhao, H., Tan, D. L., Fu, C. C., Pedrozo, V. B., Zhang, Z. Q., 2024. Improvement of the light-load combustion control strategy for a heavy-duty diesel engine fueled with diesel/methanol by RSM-NSGAIII. *Energy*, 297, 131190. https://doi.org/10.1016/j.energy.2024.131190

[7] Hoang, A. T., Foley, A. M., Nižetić, S., Huang, Z., Ong, H. C., Olçer, A. I., Pham, V. V., Nguyen, X. P., 2022. Energy-related approach for reduction of CO2 emissions: A critical strategy on the port-to-ship pathway. *Journal of Cleaner Production*, 355, 131772. https://doi.org/10.1016/j.jclepro.2022.131772

[8] Wang, J. Y., Ma, Q. Y., Nie, J. L., Lin, J. W., Su, W. H., Wu, B. Y., 2025. Influence mechanism and optimization of the diesel engine jet disturbance chamber injection strategy for enhanced combustion and thermal efficiency. *Applied Thermal Engineering*, 261, 125169. https://doi.org/10.1016/j.applthermaleng.2024.125169

[9] Dai, S., Chen, Q., Dai, H., 2019. Prediction of Exhaust Gas Temperature Margin Based on LSSVR. *2019 Chinese Control Conference (CCC)*, 27-30 July, Guangzhou, China, 6808-6812. https://doi.org/10.23919/ChiCC.2019.8866175

[10] Knežević, V., Orović, J., Stazić, L., Čulin, J., 2020. Fault Tree Analysis and Failure Diagnosis of Marine Diesel Engine Turbocharger System. *Journal of Marine Science and Engineering*, 8(12), 1004. https://doi.org/10.3390/jmse8121004

[11] Puzdrowska, P., 2023. Diagnostic Analysis of Exhaust Gas with A Quick-Changing Temperature from a Marine Diesel Engine Part II / Two Factor Analysis. *Polish Maritime Research*, 30(3), 89-95. https://doi.org/10.2478/pomr-2023-0042

[12] Zhou, Z., Bao, T., Ding, J., Chen, Y., Wang, F., Zhang, B., 2024. Diesel Engine Monitoring and Diagnostics Based on Artificial Neural Networks. *2024 13th International Conference on Communications, Circuits and Systems (ICCCAS)*, 10-12 May, Xiamen, China, 131-135. https://doi.org/10.1109/ICCCAS62034.2024.10652796

[13] Nguyen, G. V., Sakthivel, R., Rudzki, K., Kozak, J., Prabhakar, S., Pham, N. D. K., Nguyen, P. Q. P., Phuong, N. X., 2023. Using artificial neural networks for predicting ship fuel consumption. *Polish Maritime Research*, 30, 39-60. https://doi.org/10.2478/pomr-2023-0020

[14] Nguyen, V. N., Chung, N., Balaji, G. N., Rudzki, K., Hoang, A. T., 2025. Internet of things-driven approach integrated with explainable machine learning models for ship fuel consumption prediction. *Alexandria Engineering Journal*, 118, 664-680. https://doi.org/10.1016/j.aej.2024.12.045

[15] Hoang, A. T., Bui, T. A. E., Nguyen, X. P., Bui, V. H., Nguyen, Q. C., Truong, T. H., Chung, N., 2025. Explainable machine learning-based prediction of fuel consumption in ship main engines using operational data. *Brodogradnja*, 76(4), 76405. https://doi.org/10.21278/brod76405

[16] Meng, L., Gan, H., Liu, H., Lu, D., 2025. Deep learning-based research on fault warning for marine dual fuel engines. *Brodogradnja*, 76(3), 76303. https://doi.org/10.21278/brod76303

[17] Liu, B., Gan, H., Chen, D., Shu, Z., 2023. Research on fault early warning of marine diesel engine based on CNN-BiGRU. *Journal of Marine Science and Engineering*, 11(1), 56. https://doi.org/10.3390/jmse11010056

[18] Ji, Z., Gan, H., Liu, B., 2023. A deep learning-based fault warning model for exhaust temperature prediction and fault warning of marine diesel engine. *Journal of Marine Science and Engineering*, 11(8), 1509. https://doi.org/10.3390/jmse11081509

[19] Tian, X., Gan, H., Sun, F., 2025. PSO–CNN–BiLSTM: a marine diesel engine fault early warning model. *Ships and Offshore Structures*. https://doi.org/10.1080/17445302.2025.2537939

[20] Sun, J., Ren, H., Duan, Y., Yang, X., Wang, D., Tang, H., 2024. Fusion of multi-layer attention mechanisms and CNN-LSTM for fault prediction in marine diesel engines. *Journal of Marine Science and Engineering*, 12(6), 990. https://doi.org/10.3390/jmse12060990

[21] Zhao, C., 2014. A quality-relevant sequential phase partition approach for regression modeling and quality prediction analysis in manufacturing processes. *IEEE Transactions on Automation Science and Engineering*, 11(4), 983-991. https://doi.org/10.1109/TASE.2013.2287347

[22] Karimi, S., Dibeklioğlu, H., 2026. Shift-ACT: Test-time adaptation via shift-aware class-specific dynamic thresholding for domain generalization. *Neurocomputing*, 663, 132013. https://doi.org/10.1016/j.neucom.2025.132013

[23] Oyucu, S., Aksöz, A., 2024. Integrating machine learning and MLOps for wind energy forecasting: a comparative analysis and optimization study on Türkiye's wind data. *Applied Sciences*, 14(9), 3725. https://doi.org/10.3390/app14093725

[24] Suárez, D., Almeida, F., Blanco, V., Toledo, P., 2025. KubePipe: a container-based high-level parallelization tool for scalable machine learning pipelines. *The Journal of Supercomputing*, 81, 451. https://doi.org/10.1007/s11227-025-06956-x

[25] Drin, S., Shchestyuk, N., 2024. Forecast model of the price of a product with a cold start. *Mathematical and Statistical Methods for Actuarial Sciences and Finance*. Springer, Cham, 154-159. https://doi.org/10.1007/978-3-031-64273-9_26

[26] Hörbe, R., Erol, S., 2025. Optimizing LightGBM for regression: a study on parameter influence and performance. *IFAC-PapersOnLine*, 59(10), 2292-2297. https://doi.org/10.1016/j.ifacol.2025.09.385

[27] Luo, Y., Xu, Q., Li, W., Jiang, F., Xiao, B., 2021. A multi-step decision prediction model based on LightGBM. *2021 IEEE International Conference on Big Data (Big Data)*, 15-18 December, Orlando, FL, USA, 5714-5718. https://doi.org/10.1109/BigData52589.2021.9671558

[28] Zhang, J., Mucs, D., Norinder, U., Svensson, F., 2019. LightGBM: an effective and scalable algorithm for prediction of chemical toxicity–application to the Tox21 and Mutagenicity data sets. *Journal of Chemical Information and Modeling*, 59(10), 4150-4158. https://doi.org/10.1021/acs.jcim.9b00633

[29] Quinlan, J. R., 1986. Induction of decision trees. Machine Learning, 1, 81-106. https://doi.org/10.1007/BF00116251

[30] Dhulipalla, R., Vincent, S., 2025. Earthquake magnitude detection: leveraging machine learning with metaheuristics for feature selection and dimensionality reduction. *Results in Earth Sciences*, 3, 100145. https://doi.org/10.1016/j.rines.2025.100145

[31] Hodson, T. O., 2022. Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. *Geoscientific Model Development*, 15, 5481-5487. https://doi.org/10.5194/gmd-15-5481-2022

[32] Fang, Z., Mengaldo, G., 2025. Dynamical errors in machine learning forecasts. *Chaos, Solitons & Fractals*, 201(3), 117376. https://doi.org/10.1016/j.chaos.2025.117376

[33] Chicco, D., Warrens, M. J., Jurman, G., 2021. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, 7, e623. https://doi.org/10.7717/peerj-cs.623

[34] Wu, H. X., Mao, Y. Q., Weng, J. C., Yu, Y., Wang, J. H., 2025. Fractional light gradient boosting machine ensemble learning model: A non-causal fractional difference descent approach. *Information Fusion*, 118, 102947. https://doi.org/10.1016/j.inffus.2025.102947

[35] Barros, F. S., Cerqueira, V., Soares, C., 2021. Empirical Study on the Impact of Different Sets of Parameters of Gradient Boosting Algorithms for Time-Series Forecasting with LightGBM. *PRICAI 2021: Trends in Artificial Intelligence*. Lecture Notes in Computer Science, 13031. *Springer*, Cham, 454-465. https://doi.org/10.1007/978-3-030-89188-6_34

[36] Woźniak, A. P., Milczarek, M., Woźniak, J., 2025. MLOps components, tools, process, and metrics: a systematic literature review. *IEEE Access*, 13, 22166-22175. https://doi.org/10.1109/ACCESS.2025.3534990

[37] Antonini, M., Pincheira, M., Vecchio, M., Antonelli, F., 2022. Tiny-MLOps: a framework for orchestrating ML applications at the far edge of IoT systems. *2022 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, 25-26 May, Larnaca, Cyprus, 1-8. https://doi.org/10.1109/EAIS51927.2022.9787703

[38] Testi, M., Ballabio, M., Frontoni, E., Iannello, G., Moccia, S., Soda, P., 2022. MLOps: a taxonomy and a methodology. *IEEE Access*, 10, 63606-63618. https://doi.org/10.1109/ACCESS.2022.3181730

[39] Rzig, D. E., Hassan, F., Bansal, C., Nagappan, N., 2022. Characterizing the Usage of CI Tools in ML Projects. *16th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 19-23 September, Helsinki, Finland, 69-79. https://doi.org/10.1145/3544902.3546237

[40] Kodakandla, N., 2024. Data drift detection and mitigation: A comprehensive MLOps approach for real-time systems. *International Journal of Science and Research Archive*, 12(01), 3127-3139. https://doi.org/10.30574/ijsra.2024.12.1.0724

[41] Faubel, L., Woudsma, T., Kloepper, B., Eichelberger, H., Buelow, F., Schmid, K., Ghezeljehmeidan, A. G., Methnani, L., Theodorou, A., Bång, M., 2025. MLOps for cyberphysical production systems: challenges and solutions. *IEEE Software*, 42(1), 65-73. https://doi.org/10.1109/MS.2024.3441101

[42] Cao, Y., Zhang, J., Ma, A., Xu, H., Liu, J., 2026. Marine engine cylinder exhaust temperature prediction based on PSO-optimized CNN-LSTM-attention network. *Brodogradnja*, 77(1), 77101. https://doi.org/10.21278/brod77101

[43] Tu, S., Li, C., Shepherd, B. E., 2025. Between- and within-cluster Spearman rank correlations. *Statistics in Medicine*, 44(3-4), e10326. https://doi.org/10.1002/sim.10326

[44] Croux, C., Dehon, C., 2010. Influence functions of the Spearman and Kendall correlation measures. *Statistical Methods & Applications*, 19, 497-515. https://doi.org/10.1007/s10260-010-0142-z

[45] Jiang, J. F., Zhang, X. Y., Yuan, Z., 2024. Feature selection for classification with Spearman's rank correlation coefficient-based self-information in divergence-based fuzzy rough sets. *Expert Systems with Applications*, 249, 123633. https://doi.org/10.1016/j.eswa.2024.123633

[46] Schober, P., Boer, C., Schwarte, L. A., 2018. Correlation coefficients: appropriate use and interpretation. *Anesthesia & Analgesia*, 126(5), 1763-1768. https://doi.org/10.1213/ANE.0000000000002864

[47] Gu, J., Wang, Y. Y., Li, J. W., Zhang, K., Shi, L., Deng, K. Y., 2024. Research on load adaptability expansion of zero-dimensional heat release prediction for direct-injection diesel engines. *Fuel*, 368, 131518. https://doi.org/10.1016/j.fuel.2024.131518

[48] Liu, Y. G., Song, Y. P., Leng, L., Shi, L., Deng, K. Y., 2024. Effects of control strategies of the electric supercharger on transient processes of a turbocharged diesel engine. *Energy*, 307, 132427. https://doi.org/10.1016/j.energy.2024.132427

[49] Serrano, J. R., Olmeda, P., Arnau, F. J., Dombrovsky, A., Smith, L., 2015. Turbocharger heat transfer and mechanical losses influence in predicting engines performance by using one-dimensional simulation codes. *Energy*, 86, 204-218. https://doi.org/10.1016/j.energy.2015.03.130

[50] Cerqueira, V., Torgo, L., Mozetič, I., 2020. Evaluating time series forecasting models: an empirical study on performance estimation methods. *Machine Learning*, 109, 1997-2028. https://doi.org/10.1007/s10994-020-05910-7

[51] Pislyakov, V., 2022. On some properties of medians, percentiles, baselines, and thresholds in empirical bibliometric analysis. *Journal of Informetrics*, 16(4), 101322. https://doi.org/10.1016/j.joi.2022.101322

[52] Ilyushin, B. B., 2024. On applicability of IQR method for filtering of experimental data. *Journal of Engineering Thermophysics*, 33, 1-8. https://doi.org/10.1134/S1810232824010016

[53] Sun, R. K., Abouarghoub, W., Demir, E., 2025. Enhancing data quality in maritime transportation: A practical method for imputing missing ship static data. *Ocean Engineering*, 315, 119722. https://doi.org/10.1016/j.oceaneng.2024.119722

[54] Zhao, J., He, J., Wang, J., Liu, K., 2025. Energy consumption prediction for electric buses based on traction modeling and LightGBM. *World Electric Vehicle Journal*, 16(3), 159. https://doi.org/10.3390/wevj16030159

[55] Barbato, G., Barini, E. M., Genta, G., Levi, R., 2011. Features and performance of some outlier detection methods. *Journal of Applied Statistics*, 38(10), 2133–2149. https://doi.org/10.1080/02664763.2010.545119

[56] Schwertman, N. C., de Silva, R., 2007. Identifying outliers with sequential fences. *Computational Statistics & Data Analysis*, 51(8), 3800-3810. https://doi.org/10.1016/j.csda.2006.01.019

[57] Castresana, J., Gabiña, G., Quincoces, I., Uriondo, Z., 2023. Healthy marine diesel engine threshold characterisation with probability density functions and ANNs. *Reliability Engineering & System Safety*, 238, 109466. https://doi.org/10.1016/j.ress.2023.109466

[58] Cui, K. X., Wu, W. J., Shang, J., Shi, D. W., 2025. Dynamic alarm monitoring with data-driven ellipsoidal threshold learning. *Control Engineering Practice*, 158, 106282. https://doi.org/10.1016/j.conengprac.2025.106282

[59] Zhao, H., Pan, R., 2025. Gaussian derivative change-point detection for early warnings of industrial system failures. *Reliability Engineering & System Safety*, 256, 110681. https://doi.org/10.1016/j.ress.2024.110681

[60] Burke, R. D., Madamedon, M., Williams, R., 2020. Newly identified effects of injector nozzle fouling in diesel engines. *Fuel*, 278, 118336. https://doi.org/10.1016/j.fuel.2020.118336

[61] von Stackelberg, P., Goedhart, R., Birbil, İ., Does, R. J. M. M., 2024. Comparison of threshold tuning methods for predictive monitoring. *Quality and Reliability Engineering International*, 40(1), 499-512. https://doi.org/10.1002/qre.3436

[62] Zeng, W., Wang, Q. H., Liu, F. L., Wang, Y., 2016. Learning from adaptive neural network output feedback control of a unicycle-type mobile robot. *ISA Transactions*, 61, 337-347. https://doi.org/10.1016/j.isatra.2016.01.005

[63] Pujatti, M., Calzà, D., Gobbi, A., Svaizer, P., Cristoforetti, M., 2025. Mopidip: a modular real-time pipeline for machinery diagnosis and prognosis based on deep learning algorithms. *Journal of Big Data*, 12, 82. https://doi.org/10.1186/s40537-025-01123-8

[64] Zhou, L., Wang, H. W., 2025. Multi-task model of adaptive multi-scale feature fusion and adaptive mixture-of-experts for equipment remaining useful life prediction and fault diagnosis. *Expert Systems with Applications*, 272, 126807. https://doi.org/10.1016/j.eswa.2025.126807